

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re U.S. Patent Application of

SUZUKI et al.

Application Number: To be Assigned

Filed: Concurrently Herewith

**For: DATABASE SYSTEM, SERVER, QUERY POSING
METHOD, AND DATA UPDATING METHOD**

ATTORNEY DOCKET NO. HITA.0477

**Honorable Assistant Commissioner
for Patents
Washington, D.C. 20231**

**REQUEST FOR PRIORITY
UNDER 35 U.S.C. § 119
AND THE INTERNATIONAL CONVENTION**

Sir:

In the matter of the above-captioned application for a United States patent, notice is hereby given that the Applicant claims the priority date of March 17, 2003, the filing date of the corresponding Japanese patent application 2003-071908.

A certified copy of Japanese patent application 2003-071908 is being submitted herewith. Acknowledgment of receipt of the certified copy is respectfully requested in due course.

Respectfully submitted,

Stanley P. Fisher
Registration Number 24,344

Juan Carlos A. Marquez
Registration Number 34,072

REED SMITH LLP
3110 Fairview Park Drive
Suite 1400
Falls Church, Virginia 22042
(703) 641-4200
January 6, 2004



日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 3 月 1 7 日
Date of Application:

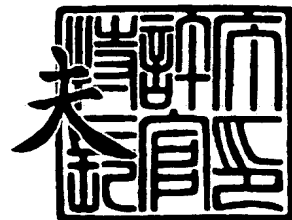
出 願 番 号 特 願 2 0 0 3 - 0 7 1 9 0 8
Application Number:
[ST. 10/C]: [J P 2 0 0 3 - 0 7 1 9 0 8]

出 願 人 株式会社日立製作所
Applicant(s):

2 0 0 3 年 9 月 2 6 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



出証番号 出証特 2 0 0 3 - 3 0 7 9 4 0 1

【書類名】 特許願

【整理番号】 GM0302009

【提出日】 平成15年 3月17日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/46
G06F 15/16

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所 中央研究所内

【氏名】 鈴木 芳生

【発明者】

【住所又は居所】 東京都国分寺市東恋ヶ窪一丁目 2 8 0 番地 株式会社日立製作所 中央研究所内

【氏名】 藤原 真二

【特許出願人】

【識別番号】 000005108

【氏名又は名称】 株式会社日立製作所

【代理人】

【識別番号】 100075513

【弁理士】

【氏名又は名称】 後藤 政喜

【選任した代理人】

【識別番号】 100084537

【弁理士】

【氏名又は名称】 松田 嘉夫

【選任した代理人】

【識別番号】 100114236

【弁理士】

【氏名又は名称】 藤井 正弘

【手数料の表示】

【予納台帳番号】 019839

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 0110326

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データベースシステム、サーバ、問い合わせ投入方法及びデータ更新方法

【特許請求の範囲】

【請求項 1】

同一内容の検索が可能なデータベースを有し、問い合わせ要求に従って該データベースを検索する複数のデータベースサーバと、

前記問い合わせ依頼を受け付け、定められたルールを用いて前記データベースサーバに問い合わせを投入するフロントエンドサーバと、

前記フロントエンドサーバが使用するルールを管理する管理サーバと、

前記各サーバ及び問い合わせを要求するクライアント端末を接続するネットワークと、を備えるデータベースシステムであって、

前記管理サーバは、

前記データベースサーバの実行ログを取得するログ取得手段と、

前記取得したログを用いて計算された問い合わせに関する相性に基づいて前記ルールを生成するルール生成手段と、を有し、

前記フロントエンドサーバは、前記管理サーバで生成されたルールを用いて、前記問い合わせを投入する問合せ投入手段を有することを特徴とするデータベースシステム。

【請求項 2】

同一内容の検索が可能なデータベースを検索する複数のデータベースサーバに対して、定められたルールを用いて、問い合わせ要求に従って問い合わせを投入するフロントエンドサーバが使用するルールを管理するサーバであって、

前記データベースサーバの実行ログを取得するログ取得手段と、

前記取得したログを用いて計算された問い合わせに関する相性に基づいて前記ルールを生成するルール生成手段と、

前記生成されたルールを用いて前記フロントエンドサーバに前記問い合わせを投入させるために、前記生成されたルールを前記フロントエンドサーバに送信するルール送信手段と、有することを特徴とするサーバ。

【請求項 3】

前記ルール生成手段は、前記取得したログに含まれる問い合わせを統計的に処理することによって前記問い合わせ間の相性を計算し、ルールを生成することを特徴とする請求項 2 に記載のサーバ。

【請求項 4】

前記ルール生成手段は、前記データベースサーバの実行ログに含まれる問い合わせをグループ化し、

グループ化された問い合わせの平均処理時間と、

該グループ化された問い合わせ及び他のグループ化された問い合わせが所定の時間関係で実行される場合の、該グループ化された問い合わせの平均処理時間と、を比較した結果に基づいて、

前記問い合わせ間の相性を計算し、ルールを生成することを特徴とする請求項 3 に記載のサーバ。

【請求項 5】

前記ルール生成手段は、

前記データベース毎に前記実行ログ内へ記録される問い合わせの処理時間を、前記データベース間で比較した結果に基づいて、

前記問い合わせと前記データベースとの間の相性を計算し、ルールを生成することを特徴とする請求項 2 に記載のサーバ。

【請求項 6】

前記ルール生成手段は、

あるユーザが実行した問い合わせの平均処理時間と、

該ユーザが実行する問い合わせ及び他のユーザが実行した問い合わせが、所定の時間関係で実行される場合の、該ユーザが投入した問い合わせの平均処理時間とを、比較した結果に基づいて、

問い合わせを発行するユーザ間の相性を計算し、ルールを生成することを特徴とする請求項 2 に記載のサーバ。

【請求項 7】

前記ルール生成手段によって生成されたルールを変更するルール変更手段を有

、 することを特徴とする請求項 2 に記載のサーバ。

【請求項 8】

ネットワークを介して外部からのアクセスを許容する外部アクセス手段を有し、

前記外部アクセス手段を介して、前記ルール変更手段によって前記ルールを変更することを特徴とする請求項 7 に記載のサーバ。

【請求項 9】

前記ログ取得手段は、前記ルール送信手段によって送信されたルールに基づいて投入された問い合わせに関する実行ログを取得し、

前記ルール生成手段は、前記取得したログを用いて計算された問い合わせに関する相性に基づいて前記ルールを生成することによって、

自律的にルールを学習することを特徴とする請求項 2 に記載のサーバ。

【請求項 10】

同一内容の検索が可能なデータベースを有し、問い合わせ要求に従って該データベースを検索する複数のデータベースサーバと、

前記問い合わせ依頼を受け付け、定められたルールを用いて前記データベースサーバに問い合わせを投入するフロントエンドサーバと、

前記フロントエンドサーバが使用するルールを管理する管理サーバと、を備えるデータベースシステムで用いられる問い合わせ投入方法において、

前記管理サーバは、

前記データベースサーバの実行ログを取得し、

前記取得したログを用いて計算された問い合わせに関する相性に基づいて前記ルールを生成し、

前記フロントエンドサーバは、

前記管理サーバで生成されたルールを用いて、前記問い合わせを投入することを特徴とする問い合わせ投入方法。

【請求項 11】

前記取得したログに含まれる問い合わせを統計的に処理することによって前記問い合わせ間の相性を計算し、ルールを生成することを特徴とする請求項 10 に

記載の問い合わせ投入方法。

【請求項 12】

前記データベースサーバの実行ログに含まれる問い合わせをグループ化し、
前記グループ化された問い合わせの平均処理時間と、
該グループ化された問い合わせ及び他のグループ化された問い合わせが所定の
時間関係で実行される場合の、該グループ化された問い合わせの平均処理時間と
、を比較した結果に基づいて、

前記問い合わせ間の相性を計算することを特徴とする請求項 11 に記載の問い
合わせ投入方法。

【請求項 13】

前記取得したログに含まれる問い合わせを統計的に処理することによって、前
記問い合わせと前記データベースサーバとの間の相性を計算し、ルールを生成す
ることを特徴とする請求項 10 に記載の問い合わせ投入方法。

【請求項 14】

前記取得したログに含まれる問い合わせを統計的に処理することによって、前
記問い合わせに関して、前記問い合わせを発行するユーザ間の相性を計算し、ル
ールを生成することを特徴とする請求項 10 に記載の問い合わせ投入方法。

【請求項 15】

前記問い合わせを投入するデータベースサーバを決定する際に、該問いあわせ
と最も相性のよいデータベースサーバを選択することを特徴とする請求項 10 に
記載の問い合わせ投入方法。

【請求項 16】

前記問い合わせを投入するデータベースサーバを決定する際に、
該問いあわせと相性のよいデータベースサーバが無い場合は、
該問い合わせを投入可能なデータベースサーバの中から、該問い合わせと相性
が悪いデータベースサーバサーバを除外した後に、
該問い合わせを投入するデータベースサーバを確率的に選択することを特徴と
する請求項 10 に記載の問い合わせ投入方法。

【請求項 17】

前記問い合わせを投入するデータベースサーバを決定する際に、
該問い合わせとの相性のよい問い合わせが直前に実行されているデータベースサーバを選択して、該問い合わせを投入することを特徴とする請求項 10 に記載の問い合わせ投入方法。

【請求項 18】

前記送信されたルールに基づいて投入された問い合わせに関する実行ログを取得し、

前記取得したログを用いて計算された問い合わせに関する相性に基づいて前記ルールを生成することによって、自律的にルールを学習することを特徴とする請求項 10 に記載の問い合わせ投入方法。

【請求項 19】

同一内容の検索が可能なデータベースを検索する複数のデータベースサーバに対して、定められたルールを用いて、問い合わせ要求に従って問い合わせを投入するフロントエンドサーバが使用するルールを管理する管理サーバのデータを更新するデータ更新方法であって、

ネットワークを介して外部から前記管理サーバにアクセスし、
前記管理サーバが保有するルールを取得して、
前記取得したルールに基づいて分析レポートを生成し、
前記分析レポートに基づいて、前記管理サーバが保有するルールを変更することを特徴とするデータ更新方法

【請求項 20】

前記分析レポートによって、データベースサーバの稼働の障害となるルールを検出し、

該検出されたルールを変更することを特徴とする請求項 19 に記載のデータ更新方法

【発明の詳細な説明】

【0001】

【発明が属する技術分野】

本発明は、問い合わせを実行する複数のデータベースサーバと、外部から受け

付けた問い合わせをデータベースサーバへ投入するフロントエンドサーバとから構成される検索システムに関する。

【0002】

【従来の技術】

近年、性能や信頼性向上を目的として、データベースの並列化・分散化が進められている。図22に、従来のデータベースシステムの典型的な構成例を示す。

【0003】

この従来のデータベースシステムは、一つのマスターデータベース103に対し、複数のレプリカデータベース104を作成し、フロントエンドサーバ102は、それらのレプリカデータベース104に問い合わせを分散して投入することによって、データベースの検索性能の向上が図られている。また、一方を運用系とし、他方を待機系として、障害時に両者を切りかえることによって信頼性の向上が図られる場合もある。

【0004】

このように複数のサーバに対して問い合わせを分散して投入する方法として、ラウンドロビンで割り当てるサーバを決定する方法、又は、CPU利用率等の負荷を計測し負荷の軽いサーバへ割り当てる方法が従来から用いられている。

【0005】

例えば、各バッチジョブのリソース使用量を計算しておき、複数のジョブを実行する場合には、累計のリソースを計算し、累計リソースがサーバの許容量を超える場合には、新たなジョブを投入しないことによってリソース競合を回避するバッチジョブのスケジューリング方法が提案されている（例えば、特許文献1参照。）。

【0006】

データベースシステムの性能をさらに向上させるためには、レプリカデータベース104における、データベースバッファ競合、ディスク競合等のリソース競合を回避することが重要となる。

【0007】

以下、図23、図24を用いてデータベースバッファ競合（キャッシュ競合）

について説明する。ディスク 201 はサーバ 200 に接続されており、ディスク 201 には 3 つの表 (205、206、207) が格納されている。表 1 (205) のデータを要求する問い合わせ 1 (220) が、サーバ 200 に投入されると、ディスク 201 上の表 1 (205) から必要なデータが問い合わせに送信される。

【0008】

ディスクの入出力 (ディスク I/O) は、メモリの入出力に比べ、多くの処理時間を要するため、ディスクの入出力のキャッシュとしてメモリが用いられる。このキャッシュ領域 (データベースバッファ) はメモリ 203 上に作成され、数 KB のデータページ 204 に分割されて構成されている。データページ 204 は LRU (Least Recently Used) によって、頻繁に利用されるデータほどメモリ 203 上に残るように管理される。

【0009】

表 1 (205) のデータを取得する問い合わせ 1 (220) が発行された場合、データベースバッファが検索され、メモリ 203 上のデータベースバッファに必要とする表 1 (205) のデータが記憶されていれば、ディスク I/O なしで結果を得ることが可能である。

【0010】

一方、図 24 (A) に示すように、表 2 (206) のデータを取得する問い合わせ 2 (221) が、サーバ 200 に投入されると、メモリ 300 に必要なデータが記憶されていないため、ディスク I/O が行われる。そして、図 24 (B) に示すようにメモリ 301 の一部は、問い合わせの内容である表 2 (206) のデータで上書きされる。

【0011】

例えば、それぞれの内容が異なり、かつ、大きな結果を要求する二つの問い合わせが連続して実行される場合、お互いがメモリ 300 上のデータベースバッファを上書きしあい、問い合わせの実行の度にディスク I/O が生じる。逆に、内容が共有できる問い合わせの場合は、キャッシュ (データベースバッファ) が記憶されているデータを使用できる可能性が高く、少ないディスク I/O で結果

を得ることができる。このように、問い合わせには相性があり、問い合わせの投入順序によって、データベースシステムの性能が変化する。

【0012】

データベースバッファ競合（キャッシュ競合）を回避する方法としては、同じデータを要求する複数の問い合わせ間でデータベースバッファを共有させる方法がある。例えば、2つの異なる問い合わせ（クエリ）が、同一のデータを要求し、かつそのデータがデータベースバッファ領域より大きい場合に、最初のクエリがデータを途中まで読んだところで、2番目の問い合わせ（クエリ2）クエリが呼ばれると、データの前半部分はすでにデータベースバッファ上から消去されている可能性がある。すなわち、クエリが必要とするデータが大きい場合には、1番目の問い合わせ（クエリ1）が現在読んでいるデータの後半部分によって、該データの前半部分が上書きされている可能性がある。

【0013】

このような場合、該データの前半部分は、クエリ2によってディスクから再度読み込みされ、バッファが上書きされる。しかし、上書きした部分に記憶されていたデータ（クエリ1が読みこんだデータの後半部分）は、この後クエリ2によっても必要とされるデータなので、無駄なディスクI/Oが生じる。そこで、2番目の読みこみでは、最初から読むのではなく、最初の問い合わせ（クエリ1）とバッファを共有しながら、実行中のクエリ1と同じ場所から読む方法（メリーゴーランドスキャン）がMicrosoft社のデータベース製品であるSQL Serverに導入されている（例えば、非特許文献1参照）。

【0014】

他のバッファを有効利用する方法としては、データの重要度に応じて、キャッシュを分類する方法もある。Oracle社のデータベース製品であるOracleでは、バッファをキープ・デフォルト・リサイクルの3つの領域に分類し、領域毎にサイズや、管理方法の指定を可能としている。例えば、バッファに常駐してほしいデータは、キープへ、上書きされてもよいものはリサイクルへ、その他はデフォルトというように、バッファを分類する。但し、設定にはデータの性質についての理解が必要であり、データのサイズやシステム構成が変化する度に設定の変更が

必要となる。

【0015】

分散化・並列化された計算機環境では、性能に加えて運用コストの増大が大きな問題となる。管理対象の計算機が増加すれば、運用コストは増大する。

【0016】

I Tシステムを用いたビジネスでは、システムダウンが莫大な損失に結びつくため、システムの安定運用が必須となる。例えば、システムへのアクセス集中によるシステムダウンを回避するために、アクセス数に応じてサーバを追加することが行われる。

【0017】

このように頻繁に構成や設定が変わる環境では、変化に追従して管理を行うことは容易ではない。環境が変化する度に、人手で計算機毎にチューニングや環境設定を実施した場合は、運用管理コストが大きくなってしまう。この運用管理コストを低下させる方法としては、自動チューニングがある。Microsoft社のSQL Serverなど、データベース単体の設定パラメータについては、自動チューニングができる製品がある。ただし、並列分散環境への対応は十分でない。

【0018】

運用コストを低下させる他の方法として、運用管理（全部又は一部）を、I Tシステムの管理を請け負うM S P（Management Service Provider）を利用して、外部に委託する方法がある。例えば、M S Pによって提供される監視サービスにおいては、サーバのC P U利用率などの性能情報を監視し、予め定めた閾値を超えたらアラームを発するといったサービスが実施される。

【0019】

【特許文献1】

特開平9-311795号公報

【非特許文献1】

”データベース アーキテクチャ”、[online]、インターネット<URL : <http://www.microsoft.com/japan/msdn/sqlserver/sql2000/thestorageengine.asp>>

【0020】**【発明が解決しようとする課題】**

データベースシステムに投入される問い合わせには、他の問い合わせやサーバとの相性がある。相性の悪い組み合わせでは、リソース競合が生じ性能が低下する。例えば、データベースバッファの競合が発生すると、メモリに比べて速度が遅いディスク I/O が発生し大きな性能上の問題が生じる。

【0021】

データベース単体でみた場合には、自動チューニング技術が開発されているが、並列・分散環境でシステム全体を最適化する技術はない。

【0022】

問い合わせを複数のサーバに分散させる方法として、例えば、ランドロビンでサーバを選択する、又は、負荷の軽いサーバを選択する方法がある。このような方法では、問い合わせ間の相性やサーバとの相性を考慮していないので、実行時にリソース競合が生じる可能性がある。そこで、並列・分散環境において、問い合わせ間や問い合わせとサーバとの間の相性を考慮して、適切なスケジューリングを実施することによって、リソース競合を回避しシステム全体のスループットを向上させることが必要となる。

【0023】

また、サーバの並列化・分散化が進み、管理対象のサーバが多くなると、個別に細かいチューニングを行うのは困難であり高コストとなる。また、単体のサーバだけでなく、システム全体として管理することは容易ではない。そこで、並列・分散環境において、スケジューリング方法を自律的に学習し、システム全体を低コストで管理することが必要となる。

【0024】**【課題を解決するための手段】**

本発明は、同一内容の検索が可能なデータベースを有し、問い合わせ要求に従って該データベースを検索する複数のデータベースサーバと、前記問い合わせ依頼を受け付け、定められたルールを用いて前記データベースサーバに問い合わせを投入するフロントエンドサーバと、前記フロントエンドサーバが使用するルー

ルを管理する管理サーバと、前記各サーバ及び問い合わせを要求するクライアント端末を接続するネットワークと、を備え、前記管理サーバは、前記データベースサーバの実行ログを取得するログ取得手段と、前記取得したログを用いて計算された問い合わせに関する相性に基づいて前記ルールを生成するルール生成手段と、を有し、前記フロントエンドサーバは、前記管理サーバで生成されたルールを用いて、前記問い合わせを投入する問合せ投入手段を有する。

【0025】

【発明の作用及び効果】

本発明では、同一内容の検索が可能なデータベースを有し、問い合わせ要求に従って該データベースを検索する複数のデータベースサーバと、前記問い合わせ依頼を受け付け、定められたルールを用いて前記データベースサーバに問い合わせを投入するフロントエンドサーバと、前記フロントエンドサーバが使用するルールを管理する管理サーバと、前記各サーバ及び問い合わせを要求するクライアント端末を接続するネットワークと、を備えるデータベースシステムで用いられる問い合わせ投入方法において、前記管理サーバは、前記データベースサーバの実行ログを取得し、前記取得したログを用いて計算された問い合わせに関する相性に基づいて前記ルールを生成し、前記フロントエンドサーバは、前記管理サーバで生成されたルールを用いて、前記問い合わせを投入する。すなわち、フロントエンドサーバにキュー及びスケジューラを設け、問い合わせの間、又は、問い合わせとデータベースサーバとの間の相性を判断し、スケジューラにおいて相性のよい組み合わせで問い合わせを投入する。そしてこのスケジューリングは、ルールに基づいて行い、リソース競合を回避するようにスケジューリングがされるので、データベースシステム全体の性能向上を図ることができる。

【0026】

また、管理サーバはデータベースサーバの実行ログを取得し、取得した実行ログを統計的に分析することによって、問い合わせ間、又は、問い合わせとデータベースサーバとの間の相性を計算し、計算した相性に基づいてルールを生成し、フロントエンドサーバに送信する。このルールは実行ログに基づいて生成されるので、環境や問い合わせの特徴が変化した場合でも自律的にルールを学習するこ

ことができる。

【0027】

【発明の実施の形態】

図1は、本発明の第1の実施の形態のデータベースシステムの構成図である。

【0028】

クライアント100は、ネットワーク101（イントラネットやインターネット（登録商標、以下同じ））を介して、データベースに対して問い合わせを発行する。

【0029】

データベースは、データベースサーバ（フロントエンド）102及びバックエンドの複数のデータベースサーバ103、104によって構成される。フロントエンド102は、問い合わせを受け付け、バックエンドのデータベースサーバ03、104に問い合わせを投入（ディスパッチ）する。

【0030】

バックエンドのデータベースサーバは、1台のマスタデータベースサーバ（マスタ）103と、複数のレプリカデータベースサーバ（レプリカ）104によって構成される。マスタ103を記憶装置106有し、レプリカ104は記憶装置107を有する。レプリカ104に接続される記憶装置107は、マスタ103に接続される記憶装置106と同じ内容が記録されている。通常、マスタ103では記憶内容の更新処理及び検索処理が行われるが、レプリカ104では検索処理のみが行われる。そして、定期的にレプリケーションを行うことによって、マスタ103の記憶装置106の変更内容をレプリカ104の記憶装置107に反映させる。

【0031】

レプリカ104は、問い合わせ実行部111及びログ記録・送信部110によって構成される。問い合わせ実行部111は、フロントエンド102が送信した問い合わせを実行する。ログ記録・送信部110は、実行のログ（問い合わせの種類、処理時間等）を記録し、管理サーバ105へ送信する。

【0032】

管理サーバ105は、ログ取得部120、相性計算部121、ルール生成部122及びルール管理部123によって構成される。また、管理サーバ105は、ルールデータベース140を有する。

【0033】

ログ取得部120は、レプリカ104が送信した実行ログを受信し、レプリカ104の実行ログを取得する。

【0034】

相性計算部121は、ログ取得部120が取得した実行ログを解析し、問い合わせに関する相性として、問い合わせ間の相性や、問い合わせとレプリカ104との相性を計算する。相性は、実行ログに記憶される問い合わせの属性間、問い合わせの属性とレプリカとの間、問い合わせを発行したユーザ間で計算することができる。

【0035】

ルール生成部122では、計算された相性に基づいて、問い合わせをスケジューリングするためのルールを生成する。例えば、問い合わせAと問い合わせBとは相性がよいので、同じレプリカ104へ投入すべきというルールが生成される。ここで、生成されたルールは、フロントエンド102へ送信される。

【0036】

ルール管理部123は、ルール生成部122によって生成されたルール及びフロントエンド102で記録された該ルールの適用回数をルールデータベース140へ格納する。また、ルールデータベース140には、各レプリカ104のハードウェアやソフトウェアの性能情報（例えば、CPU利用率、メモリ使用率等のリソース使用率）を保存することもできる。管理者は、ルール及び該ルールの適用回数に関する情報によって実際にどのルールが使用されているかを管理することが可能になる。

【0037】

フロントエンド102は、キュー131及びスケジューラ130を有し、クライアント100から受信した問い合わせをキュー131で受け付け、スケジューラ130がスケジューリングして、問い合わせ104を分配して、投入する。

【0038】

スケジューラ130は、管理サーバ105が生成したルールに基づいてスケジューリングをし、問い合わせを投入するレプリカ104を決定する。レプリカ104への問い合わせの投入の際、ルール毎にそのルールが適用された回数を記録しておく。ルールの適用回数に関する情報は、管理サーバ105へ送信される。

【0039】

レプリカ104が問い合わせを実行する。そして、レプリカ104において実行された問い合わせに基づいて、管理サーバ105が相性を計算し、ルールを生成する。そして、管理サーバにおいて生成されたルールに基づいて、フロントエンド102がレプリカ104に問い合わせを投入する。そして、レプリカ104が問い合わせを実行する。

【0040】

このサイクルが繰り返されることによって、システム変更（例えば、サーバの追加、データサイズの増加）や、入力の変化（例えば、入力負荷の増加、問い合わせ内容の変化）に適応したスケジューリングルールの自律的な生成が可能になる。フロントエンド102において、生成されたスケジューリングルールを用いて、リソースの競合を回避する組み合わせで問い合わせを投入することによって、データベースシステム全体の性能向上を低コストで実現することが可能となる。

【0041】

次に、レプリカ104、管理サーバ105及びフロントエンド102の処理について説明する。

【0042】

図2は、本発明の第1の実施の形態のデータベースサーバ（レプリカ104）の処理を示すフローチャートである。

【0043】

まず、クライアント100からの問い合わせを受け付ける（ステップ400）。そして、受け付けた問い合わせを実行する（ステップ401）。続いて、問い合わせの実行ログ410を記録し（ステップ402）、実行ログ410を管理サ

サーバ 105 へ送信する。そして、次の問い合わせを受け付ける（ステップ 400）。

【0044】

実行ログ 410 は、問い合わせ単位でリアルタイムに送信してもよい。例えば、問い合わせの実行毎にイベントを送信し、実行ログの取得先（管理サーバ 105）では、イベントを取得することによって、実行ログを記録する（例えば、Microsoft 社の SQL Server2000）。また、複数の問い合わせの実行ログ 410 をまとめて送信してもよい。例えば、一定期間、レプリカ 104 で実行ログ 410 を記録・保存しておき、適当なタイミングで管理サーバ 105 にバッチ転送する。

【0045】

実行ログ 410 は、図 3 に示す形式で送信される。すなわち、問い合わせ毎に、問い合わせ内容（SQL 文やストアドプロシージャ）700、処理時間 701、開始時刻 702、終了時刻 703、ユーザ名 704 等の問い合わせの属性を記録する。

【0046】

また、実行ログ 410 に加えて、ハードウェアやソフトウェアのリソース利用率など、他の性能情報を管理サーバ 105 へ送ることもできる。例えば、レプリカ 104 の CPU 利用率、メモリ使用率などが考えられる。これらの性能情報は、相性計算に必須ではないが、時系列データとして管理しておくことによって、レプリカ 104 の性能の監視・分析に役立つ。

【0047】

図 4 は、本発明の第 1 の実施の形態の管理サーバ 105 の処理を示すフローチャートである。

【0048】

レプリカ 104 から実行ログ 410 を取得する（ステップ 500）。

【0049】

そして、相性計算の前処理として問い合わせのグループ化を行う（ステップ 501）。相性計算は、個々の問い合わせ毎ではなく、複数の問い合わせをパターン毎に分類してグループ化したグループ毎に行う必要がある。個々の問い合わせ

毎に行うと、問い合わせの種類だけルールが生成され、該ルールは全く同一の問い合わせに対してのみ適用されるため、ルールの利用性が低くなってしまう。また、個別に計算すると、相性の組み合わせ数が多くなり、処理負担が多くなる。

【0050】

そして、グループ化の後、グループ単位に相性を計算し（ステップ502）、相性計算の結果をルール化する（ステップ503）。最後に、生成したルール510をフロントエンド102へ送信する（ステップ504）。

【0051】

フロントエンド102では、受信したルールに基づいてスケジューリングを実行して、適当なタイミング（予め定めた間隔、又は、管理サーバ105より要求があったとき）に、ルールと適用回数を管理サーバ105へフィードバックする。

【0052】

管理サーバ105では、ルール及びルールの適用回数を受信し（ステップ505）、受信した情報をルールデータベース140へ保存する（ステップ506）。

【0053】

図5は、本発明の第1の実施の形態の問い合わせのグループ化（ステップ501）を説明する図である。

【0054】

以下、“select a from A where b > 3”という単純な検索SQLを用いてグループ化について説明する。ここでは、from句の引数として記述されるAが、テーブル名である。テーブルは複数の行より構成され、行は複数の列より構成される。where句の引数では、検索条件として列名を指定する。この場合は、テーブルAの列bが3より大きいという条件を指定している。どの列を結果として返すかは、select句の引数で指定する（この例では、列aのみを取得している）。

【0055】

図5（A）に示すグループ化例1では、使用するテーブルによって、問い合わせをグループ化する。検索条件内で使用されるテーブルが同一である問い合わせ

が同一クエリにグループ化される。よって、テーブルAのみを使用する問い合わせ1011及び問い合わせ1012は、同じグループに分類され、テーブルBのみを使用する問い合わせ1013及び問い合わせ1014は、同じグループに分類される。

【0056】

図5（B）に示すグループ化例2では、テーブル名及び取得列名によって問い合わせをグループ化する。検索条件内で使用されるテーブル及び検索結果として返される列名が同一である問い合わせが同一クエリにグループ化される。よって、テーブルAの列aを取得する問い合わせ1021及び問い合わせ1022は同じグループに分類され、テーブルAの列a及びcを取得する問い合わせ1023及び問い合わせ1024は、同じグループに分類される。このとき検索条件は異なってもよい。

【0057】

図5（C）に示すグループ化例3では、テーブル名、取得列名及び検索条件によってグループ化する。検索条件内の定数値以外がすべて同一である問い合わせが同一クエリにグループ化される。よって、テーブルAの列bを用いて条件を指定し、テーブルAの列aを取得する問い合わせ1031と問い合わせ1032は同じグループに分類され、テーブルAの列cを用いて条件を指定し、テーブルAの列aを取得する問い合わせ1033と問い合わせ1034は同じグループに分類される。

【0058】

なお、図5（A）、図5（B）、図5（C）の順で詳細なグループ分けが可能になるが、グループ化のための処理量も増加する。

【0059】

次に、相性計算（ステップ502）について説明する。

【0060】

ここで、相性として、二つの問い合わせを同時に実行した場合の処理時間を、個別に実行する場合の処理時間と比べた処理時間の改善度を考える。例えば、二つの問い合わせについて検索条件内で使用されるテーブルが同じで、データベ-

スバッファが共有できる場合には、これらの二つの問い合わせを同時又は連続して実行すると、処理時間を短縮することができるので、相性がよいということになる。逆に、データベースバッファが共有できない場合は、ディスク I/O によって性能が劣化するため相性が悪いということになる。

【0061】

図6は、本発明の第1の実施の形態の問い合わせの実行状態を説明する図である。

【0062】

図6に示す例では、 $Q_i(900)$ と同じ時間に $Q_2(902)$ が実行されている。また、 $Q_1(901)$ の実行時間の一部は $Q_i(900)$ の実行時間と重複している。

【0063】

本実施の形態では、 $Q_i(900)$ の実行と重複する問い合わせ Q_j を検索する。この検索は、 Q_j の終了時間 $>$ $Q_i(900)$ の開始時間、かつ、 Q_j の開始時間 $<$ $Q_i(900)$ の終了時間、の条件を満たす Q_j を検索する。図6に示す例では、 $Q_1(901)$ 、 $Q_2(902)$ がこの条件に適合する。

【0064】

また、完全に重複する問い合わせだけを検索するのではなく、 Q_i の実行時間の前後に余裕時間 Δ を加えた時間(914)と重複する問い合わせ Q_j を検索してもよい。この検索は、 Q_j の終了時間 $>$ $Q_i(900)$ の開始時間 $-\Delta$ (912)、かつ、 Q_j の開始時間 $<$ $Q_i(900)$ の終了時間 $+\Delta$ (913)、の条件を満たす Q_j を検索する。このように、余裕時間 Δ を用いて検索をすることによって、同時に実行された問い合わせに加え、 Q_i の直前直後に実行される問い合わせも対象に含めることができる。図6に示す例では、 $Q_1(901)$ 、 $Q_2(902)$ に加えて、 $Q_3(903)$ がこの条件に適合する。

【0065】

図7は、本発明の第1の実施の形態において相性計算に用いられる相性マトリックスを説明する図である。

【0066】

相性計算は、問い合わせ間の相性を示す相性マトリックスに基づいて行う。例

例えば、相性マトリックス 800 の $C_{ij}(803)$ は、問い合わせ $Q_i(801)$ と問い合わせ $Q_j(802)$ との相性を表し、 $Q_j(802)$ と同時に実行したときの、 $Q_i(801)$ の処理時間の合計値である。一方、 $A_i(804)$ は、全ての $Q_j(802)$ との組み合わせにおける $Q_i(801)$ の処理時間の合計値を表す。

【0067】

呼出マトリックス 810 は、相性を計算するのに何個の問い合わせを使用したかを表す。例えば、 $T_{ij}(813)$ は、 $C_{ij}(803)$ を計算するのに何個の問い合わせを使用したのか（実行ログ中で、 $Q_i(801)$ が $Q_j(802)$ と同時に呼び出された回数）を表す。 $T_i(814)$ は、 $A_i(804)$ の計算に使用された問い合わせの数（実行ログに含まれる $Q_i(801)$ の実行回数）を表す。

【0068】

図 8 は、本発明の第 1 の実施の形態の相性計算（ステップ 502）を示すフローチャートであり、レプリカ 104 のログ毎に実行される。

【0069】

まず、 $Q_i(900)$ と同時に実行される問い合わせを検索する（ステップ 1101）。例えば、図 6 において説明した方法によって検索が行われる。

【0070】

検索条件に適合した全ての j について、式（1）～（3）の計算を行い、相性マトリックス 800 及び呼出マトリックス 810 の値を更新する（ステップ 1102、1103）。

【0071】

$$C_{ij} = C_{ij} + Q_i \text{ の処理時間} \quad (1)$$

$$A_i = A_i + Q_i \text{ の処理時間} \quad (2)$$

$$T_{ij} = T_{ij} + 1 \quad (3)$$

最後に、相性マトリックス 800 の全要素について、式（4）によって値を更新する（ステップ 1104、1105）。

【0072】

$$C_{ij} = A_i / T_i - C_{ij} / T_{ij} \quad (4)$$

この式（4）の右辺は、（ $Q_i(801)$ の平均処理時間）－（ $Q_j(802)$ と同時に実

行した場合の $Q_i(801)$ の平均処理時間)を表す。つまり、 $Q_i(801)$ に関して、 $Q_j(802)$ と同時に実行することによる性能改善効果が計算される。この値が $Q_i(801)$ と $Q_j(802)$ との間の相性となる。

【0073】

次に、ルールの生成(ステップ503)について説明する。

【0074】

ルールの生成(ステップ503)では、実行ログ410に記録される属性やサーバに関する相性について、ルール化を行う。例えば、問い合わせ間の相性、問い合わせとサーバとの間の相性、ユーザとサーバとの間の相性などが考えられる。ここでは、例として、問い合わせ間の相性に関するルール、問い合わせとサーバの相性に関するルールの生成について説明する。

【0075】

図9(A)は、本発明の第1の実施の形態の問い合わせ間のルールの生成(ステップ503)を示すフローチャートである。

【0076】

既に、レプリカ104毎に、相性マトリックス800、呼出マトリックス810の計算が終了している。また、図9(A)に示す計算は、レプリカ104の実行ログ410毎に行われる。

【0077】

最初に、全ての C_{ij} についてルール化の対象となるか否かを判定し、ルール化の対象を絞り込む(ステップ1201)。具体的には、 C_{ij} の絶対値を A_i で除した値と、予め定めた定数との比較結果(予め定めた定数 P_1 ($0 \leq P_1 < 1$)より大きいかな否かの判定結果)に基づいて、 C_{ij} をルール化の対象とするかを判定する。この条件による判定で、性能への影響が小さい C_{ij} をルール化への対象外とする。すなわち、相性の良い悪いにかかわらず、性能への影響の大きいものだけをルール化の対象とする。なお、 $P_1 = 0$ とした場合は、全ての C_{ij} がルール化の対象となる。

【0078】

そして、 $|C_{ij}|/A_i$ が P_1 より大きければルール化を行い(ステップ12

・ 02)、ルールリスト1へ追加する(ステップ1203)。

【0079】

図10(A)に、ルールリスト1(1910)を示す。ルールリスト1(1910)では、ルールは、問い合わせの組の条件1911、相性値1912及び回数1913によって構成される。回数1913はルールが適用された回数を表し、フロントエンド102で設定された値であり、初期値が0を設定される。

【0080】

ルールリスト1(1910)は、条件1201を満たす複数のルールによって構成される。すなわち、ルール化とは、 $C_{i,j}$ について、問い合わせ i と j の組1911として、 $C_{i,j}$ に対する相性値1912と、回数1913(初期値=0)をルールリスト1へ追加することである。

【0081】

図9(B)は、問い合わせとサーバとの間のルールの生成(ステップ503)を示すフローチャートである。

【0082】

最初に、ステップ1220において、サーバ毎に計算された Q_i の平均実行時間 $A_{i,s}$ の平均 $Ave(A_{i,s})$ を計算する(s はサーバ名)。次に、全ての $A_{i,s}$ について、条件1222において、改善率 $|A_{i,s} - Ave(A_{i,s})| / Ave(A_{i,s})$ が予め定めた定数 P_2 ($0 \leq P_2 < 1$) より大きいかな否かを判定する。この条件による判定で、性能への影響が小さい $A_{i,s}$ をルール化への対象外とする。すなわち、相性の良い悪いにかかわらず、性能への影響の大きいものだけをルール化の対象とする。なお、 $P_2 = 0$ とした場合は、全ての $A_{i,s}$ がルール化の対象となる。

【0083】

そして、 $|A_{i,s} - Ave(A_{i,s})| / Ave(A_{i,s})$ が P_2 より大きければルール化を行い(ステップ1223)、ルールリスト2に記録する(ステップ1224)。

【0084】

図10(B)は、ルールリスト2(1930)を示す。ルールリスト2(19

30) では、ルールは、問い合わせとサーバの組 1931、相性値 $(Ave(A_{i,s}) - A_{i,s})$ 1932 及び回数 1933 によって構成される。回数 1934 はルールが適用された回数を表し、フロントエンド 102 で設定された値であり、初期値が 0 に設定される。

【0085】

ルールリスト 2 (1930) は、条件 1222 を満たす複数のルールによって構成される。すなわち、ルール化とは、 $A_{i,s}$ について、問い合わせ i とサーバ s の組 1931 として、 $(Ave(A_{i,s}) - A_{i,s})$ を相性値 1932 として、回数 1933 (初期値 = 0) をルールリスト 2 へ追加することである。

【0086】

図 11 は、本発明の第 1 の実施の形態において、問い合わせとサーバ (レプリカ 104) との間の相性計算に用いられる相性マトリックスを説明する図である。

【0087】

図 7 に示す相性マトリックス 800 及び呼出マトリックス 810 は、レプリカ 104 毎に設けられている。相性マトリックス 800 の、 $A_{im}(804)$ は、レプリカ m において、全ての $Q_j(802)$ との組み合わせにおける $Q_i(801)$ の処理時間の合計値である。また、呼出マトリックス 810 の、 $T_{im}(814)$ は、レプリカ m において、 $A_i(804)$ の計算に使用された問い合わせの数 (ログ中の $Q_i(801)$ の実行回数) である。

【0088】

そして、問い合わせ $Q_i(801)$ が投入されるとき、レプリカ m についての、 $A_{im}(804)$ を比較して、 A_{im} の最小値を与えるレプリカが、 Q_i を実行するのに最も適するレプリカであるとして、レプリカと問い合わせ間の相性を判定する。

【0089】

図 12 は、本発明の第 1 の実施の形態の管理サーバ 105 が、ルールデータベース 140 において管理するデータファイルを説明する図である。

【0090】

データファイル 2004 は、ルール 2000、2001、実行ログ 2003 及

び性能情報 2002 によって構成される。ルールは、問い合わせ間の相性を表すルールリスト 1 (2000) と問い合わせとサーバの相性を表すルールリスト 2 (2001) によって構成される。ルールリスト 1 (2000)、実行ログ 2003、性能情報 2002 は、各々レプリカ 104 の数だけ存在する。

【0091】

実行ログ 2003 は、各レプリカ 104 で実行された問い合わせに関する性能情報で、図 3 において前述した形式で保存される。この実行ログ 2003 に基づいてルール 2000、2001 が生成される。

【0092】

性能情報 2004 は、各レプリカ 104 のハードウェアやソフトウェアに関する性能情報の時系列データであり、例えば、CPU 利用率、メモリ使用率等が記録されている。

【0093】

これらのデータファイル 2004 は、最新のデータだけ保存してもよいし、変更がある度に新たなデータを保存し、時系列データとして管理してもよい。

【0094】

次に、フロントエンド 102 の処理について説明する。

【0095】

図 13 は、本発明の第 1 の実施の形態のフロントエンド 102 の処理を示すフローチャートである。

【0096】

フロントエンド 102 では、受信したルールに基づいてスケジューリングを実行する。まず、管理サーバ 105 が送信したルール 510 (ルールリスト 1、ルールリスト 2) を受信する (ステップ 600)。

【0097】

そして、ルールを編集する必要があるか否かを判定し (ステップ 601)、必要な場合には編集を実行する (ステップ 602)。ルールは単純な if-then 型で記述されているので、人間 (管理者) が理解することができる。例えば、ある問い合わせは、特定のサーバで実行してほしい等の優先すべきルールが予め分かっ

“ている場合には、ルールを追加することが可能となる。また、不要なルールを削除したり、相性値を編集することによって、特定のルールが優先的に選択されるような設定をすることもできる。

【0098】

クライアント100より問い合わせを受け付けると（ステップ603）、スケジューリングを実施し（ステップ604）、レプリカ104へ問い合わせを投入する。その際、該ルールが適用された回数を、ルール毎にルールリストの回数部1913、1933（図10参照）へ記録する。そして、一定時間の経過や管理サーバ105からの要求が検出されるか否かを判定し（ステップ605）。ステップ605の条件が満たされるまでスケジューリング（ステップ604）が繰り返される。

【0099】

ステップ605の条件が満たされたタイミングで、ルールリスト520（ルール及びルールの適用回数）を管理サーバへ送信する。

【0100】

図14は、本発明の第1の実施の形態のスケジューリング（ステップ604）を示すフローチャートである。

【0101】

ここで、各レプリカ毎の問い合わせの実行数を記録する変数として“投入数”を用意する。各レプリカ104毎に投入された問い合わせ数を記録しておき、投入数が所定の閾値を超えないように制御することによって、特定のレプリカ104に問い合わせが集中して投入されることを防ぐ意味がある。

【0102】

また、問い合わせを投入可能なレプリカの組として“サーバリスト”を用意する。あるレプリカ104に対する投入数が閾値を超えた場合は、そのレプリカをサーバリストから除外し、以降の処理で、該サーバの選択を防止する。

【0103】

さらに、レプリカ毎に直前に実行したクエリを記録しておく変数として“直前クエリ”を用意する。

【0104】

スケジューリングでは、まず最初に初期化処理を行う。各レプリカの投入数を 0 に初期化し、全レプリカを含むサーバリストを構成する（ステップ 1300）。そして、全レプリカの“直前クエリ”を空欄に初期化する（ステップ 1320）。

【0105】

続いて、キューが空白か否かを判定する（ステップ 1301）。キューが空白であれば、ステップ 1301 に戻り、キューが空白でなくなるまで待機する。一方、キューが空白でなければ、サーバリストに全サーバを加えた後、レプリカ毎に投入数を判定し、投入数が予め定めた閾値（N）を超える場合は、そのレプリカに問い合わせが集中して投入されているので、サーバリストから該レプリカを除外する（ステップ 1303）。

【0106】

そして、キューから問い合わせを取りだし、ルールリスト 1 及びルールリスト 2 とのマッチングを行う（ステップ 1304）。例えば、取り出された問い合わせが Q_i である場合は、 $\{Q_i, *\}$ （* は任意）の形式のルールとマッチングを行う。

【0107】

ルールリスト 2 を用いると、* の部分がレプリカ名（ S_j ）であるルールとマッチして、 Q_i と相性が良い又は相性の悪いレプリカに関するルールが抽出される。

【0108】

ルールリスト 1 を用いると、投入可能な全てのレプリカ毎にマッチングが必要となる。最初に、“直前クエリ”変数を調査することによって、あるレプリカ（ R_1 ）の直前に投入された問い合わせ（ Q_j ）を調べ、レプリカ（ R_1 ）のルールリスト 1 に $\{Q_i, Q_j\}$ に関するルールがあるか否かを検索する。マッチするルールがあり、相性値が正であれば、レプリカ（ R_1 ）に関しては、 Q_i と Q_j との相性はよいので、 Q_i はレプリカ（ R_1 ）に投入すべきである。レプリカに対して一度も問い合わせが実行されておらず、直前の問い合わせがない場合には、

・ “ルールリスト 1 を用いたマッチングは行わない。

【0109】

次に、相性が正であるルールがマッチしたか否かを判定する（ステップ 1305）。そして、相性が正であるルールがある場合にはステップ 1306 へ進み、相性が正であるルールがない場合にはステップ 1308 へ進む。

【0110】

ステップ 1305 において相性が正であるルールがある場合には、複数のルールの中から一つのルールを選択し、そのルールが指示するレプリカへ問い合わせを投入する。選択に際しては、最も相性値の大きいルールを選択する（ステップ 1306）。なお、最も相性値の大きいルールではなく、選択されたルールの中から確率的にルールを選択して、問い合わせを投入するレプリカ決定してもよい。

【0111】

そして、投入したレプリカに対して投入数を更新（1 を加算）し、適用したルールの回数を更新（1 を加算）する（ステップ 1307）。そして、該レプリカの“直前クエリ”として、該ルールが指示する問い合わせを設定して（ステップ 1326）、ステップ 1301 へ戻る。

【0112】

一方、ステップ 1305 において相性が正であるルールがなかった場合には、相性が負であるルールがあるか否かを判定する（条件 1308）。そして、相性が負であるルールがある場合にはステップ 1309 へ進み、相性が負であるルールがない場合にはステップ 1311 へ進む。

【0113】

ステップ 1308 において相性が負であるルールがある場合には、サーバリストに含まれるレプリカから、相性が負であるルールが指示するレプリカを除外したサーバリストを作成する。そして、作成したサーバリストの中から確率的に（ランダムに）レプリカを選択し、問い合わせを投入する（ステップ 1309）。続いて、投入したレプリカに対して投入数を更新（1 を加算）し、適用したルールの回数を更新（1 を加算）する（ステップ 1310）。そして、該レプリカの

” “直前クエリ”として、該ルールが指示する問い合わせを設定して（ステップ 1323）、条件 1301 へ戻る。

【0114】

一方、ステップ 1308 において相性が負であるルールがなかった場合には、サーバリストの中から確率的に（ランダムに）サーバを選択し、問い合わせを投入する。続いて、投入したレプリカに対して投入数を更新（1を加算）し、適用したルールの回数を更新（1を加算）する（ステップ 1312）。そして、該レプリカの“直前クエリ”として、該ルールが指示する問い合わせを設定して（ステップ 1322）、条件 1301 へ戻る。

【0115】

以上説明したように、第 1 の実施の形態では、問い合わせを実行する複数のレプリカ 104 と、レプリカ 104 に問い合わせを配分して投入するフロントエンド 102 からなるデータベースシステムにおいて、フロントエンド 102 にキュー 131 及びスケジューラ 130 を設け、問い合わせの間、又は、問い合わせとレプリカ 104 との間の相性を判断し、スケジューラ 130 において相性のよい組み合わせを抱き合わせて、レプリカ 104 に問い合わせを投入する。スケジューラ 130 は、ルールに基づいて機能し、リソース競合を回避するようにスケジューリングすることによって、リソースの競合が回避され、システム全体の性能（スループット）を向上させることができる。

【0116】

また、管理サーバ 105 は、レプリカ 104 の実行ログを収集し、収集した実行ログを統計解析することによって、問い合わせ間、又は、問い合わせとレプリカとの間の相性を計算する。計算した相性に基づいてルールを生成し、フロントエンド 102 に送信する。ルールは実行ログを用いて計算されるため、システム構成、問い合わせの内容や頻度が変化した場合でも自律的に学習することが可能であり、低コストな運用が可能となる。

【0117】

また、相性は人間にも理解可能な形式でルール化されるため、ルールの編集、追加、削除が可能となり、自律的学習に人間の意思を反映させることができる。

【0118】

次に、本発明の第2の実施の形態について説明する、第2の実施の形態では、第1の実施の形態における問い合わせ間及び問い合わせとサーバ間の相性に加えて、ユーザ間の相性による判定をすることもできる。その場合は、問い合わせ間の相性マトリックス800に加え、ユーザ間の相性マトリックス2100、2101を用いる。

【0119】

図15に、本発明の第2の実施の形態において相性計算に用いられるユーザ間の相性マトリックスを説明する図である。

【0120】

相性マトリックス2100の C_{ij} (2103) は、ユーザ i (U_i : 2101) とユーザ j (U_j : 2102) との相性であり、 U_j (2102) が投入した任意のクエリと同時に実行したときの、 U_i (2101) が投入した任意の問い合わせの処理時間の合計値を表す。また、 A_i (2104) は、 U_j にかかわらず、 U_i (2101) が投入した全ての問い合わせの処理時間の合計値を表す。

【0121】

呼出マトリックス2110は、相性を計算するのに何個の問い合わせを使用したかを表す。例えば、 T_{ij} (2113) は、 C_{ij} (2103) を計算するのに何個の問い合わせを使用したのかを表す。すなわち、ログ中で、 U_i (2101) が投入した問い合わせが、 U_j (2102) が投入した問い合わせと同時に実行された回数である。 T_i (2114) は、 A_i (2104) の計算に使用された問い合わせの数 (U_i (2101) が問い合わせを実行した回数) を表す。

【0122】

問い合わせの相性と同様の方法で、ユーザ間の相性を計算し、ルールを生成し、ルールリスト3 (2130) へルールを保存する。ルールリスト3 (2130) では、ルールは、ユーザの組2131と相性値2131と回数2134によって構成される。

【0123】

図16は、本発明の第2の実施の形態のユーザ間の相性を考慮する場合のスケ

“ジューリング（ステップ604）を示すフローチャートである。なお、図16に示すスケジューリング処理は、図14に示すスケジューリング処理とステップ2200、2201、2202、2202、2203、2204において相違する。

【0124】

ここで、各レプリカ毎の問い合わせの実行数を記録する変数として“投入数”を、問い合わせを投入可能なレプリカの組として“サーバリスト”を、レプリカ毎に直前に実行したクエリを記録しておく変数として“直前クエリ”を用意する。さらに、該問い合わせを実行したユーザを記録する変数である“直前ユーザ”を用意する。

【0125】

スケジューリングでは、まず最初に初期化処理を行う。各レプリカの投入数を0に初期化し、全レプリカを含むサーバリストを構成する（ステップ1300）。そして、全レプリカの“直前クエリ”及び“直前ユーザ”を空欄に初期化する（ステップ2200）。

【0126】

続いて、キューが空白か否かを判定する（ステップ1301）。キューが空白であれば、ステップ1301に戻り、キューが空白でなくなるまで待機する。一方、キューが空白でなければ、サーバリストに全サーバを加えた後、レプリカ毎に投入数を判定し、投入数が予め定めた閾値（N）を超える場合は、そのレプリカに問い合わせが集中して投入されているので、サーバリストから該レプリカを除外する（ステップ1303）。

【0127】

そして、キューから問い合わせを取りだし、ルールリスト1及びルールリスト2とのマッチングを行う（ステップ1304）。例えば、キューから取り出した問い合わせを投入したユーザ（ U_i とする）についてマッチングを行う。このマッチングにおいては、レプリカ毎にルールリスト3に条件部が $\{U_i$, “直前ユーザ” $\}$ となるルールがあるかを調べる。すなわち、ユーザ（ U_i ）が投入する問い合わせと相性が良い問い合わせを投入するユーザがいるか、又は、相性が悪

い問い合わせを投入するユーザがいるかを判定する。

【0128】

次に、相性が正であるルールがマッチしたか否かを判定する（ステップ1305）。そして、相性が正であるルールがある場合にはステップ1306へ進み、相性が正であるルールがない場合にはステップ1308へ進む。

【0129】

ステップ1305において相性が正であるルールがある場合には、複数のルールの中から一つのルールを選択し、そのルールが指示するレプリカへ問い合わせを投入する。選択に際しては、最も相性値の大きいルールを選択する（ステップ1306）。なお、最も相性値の大きいルールではなく、選択されたルールの中から確率的にルールを選択して、問い合わせを投入するレプリカ決定してもよい。

【0130】

そして、投入したレプリカに対して投入数を更新（1を加算）し、適用したルールの回数を更新（1を加算）する（ステップ1307）。そして、該レプリカの“直前クエリ”として、該ルールが指示する問い合わせを設定し、“直前ユーザ”として、該問い合わせを投入したユーザを設定して（ステップ2204）、ステップ1301へ戻る。

【0131】

一方、ステップ1305において相性が正であるルールがなかった場合には、相性が負であるルールがあるか否かを判定する（条件1308）。そして、相性が負であるルールがある場合にはステップ1309へ進み、相性が負であるルールがない場合にはステップ1311へ進む。

【0132】

ステップ1308において相性が負であるルールがある場合には、サーバリストに含まれるレプリカから、相性が負であるルールが指示するレプリカを除外したサーバリストを作成する。そして、作成したサーバリストの中から確率的に（ランダムに）レプリカを選択し、問い合わせを投入する（ステップ1309）。続いて、投入したレプリカに対して投入数を更新（1を加算）し、適用したルー

ルの回数を更新（1を加算）する（ステップ1310）。そして、該レプリカの“直前クエリ”として、該ルールが指示する問い合わせを設定し、“直前ユーザ”として、該問い合わせを投入したユーザを設定して（ステップ2203）、ステップ1301へ戻る。

【0133】

一方、ステップ1308において相性が負であるルールがなかった場合には、サーバリストの中から確率的に（ランダムに）サーバを選択し、問い合わせを投入する。続いて、投入したレプリカに対して投入数を更新（1を加算）し、適用したルールの回数を更新（1を加算）する（ステップ1312）。そして、該レプリカの“直前クエリ”として、該ルールが指示する問い合わせを設定し、“直前ユーザ”として、該問い合わせを投入したユーザを設定して（ステップ2202）、ステップ1301へ戻る。

【0134】

以上説明したように、第2の実施の形態では、問い合わせを実行する複数のレプリカ104と、レプリカ104に問い合わせを配分して投入するフロントエンド102からなるデータベースシステムにおいて、フロントエンド102にキュー131及びスケジューラ130を設け、問い合わせを投入するユーザ間の相性、又は、問い合わせを投入するユーザとレプリカ104との相性を判断し、スケジューラ130において相性のよい組み合わせを抱き合わせて、レプリカ104に問い合わせを投入する。スケジューラ130は、ルールに基づいて機能し、リソース競合を回避するようにスケジューリングすることによって、リソースの競合が回避され、システム全体の性能（スループット）を向上させることができる。

【0135】

次に、本発明の第3の実施の形態について説明する、第3の実施の形態においては、外部サーバをネットワーク経由で管理サーバに接続し、外部サーバから管理サーバへのアクセスを可能とすることによって、外部から監視、管理を行うことができる。

【0136】

図 17 は、本発明の第 3 の実施の形態のデータベースシステムの構成図である。

【0137】

管理サーバ 105 はネットワーク 1601 (イントラネットやインターネット) に接続される。管理サーバ 105 はネットワーク 1601 を介して、管理を請け負う MSP (Management Service Provider) の監視・分析サーバへ接続 1600 される。管理サーバ 105 のルールデータベース 140 には、ルールや性能データが蓄積されており、監視・分析サーバ 1600 でそれらを分析することによって、外部からのリモート保守やシステムの診断をすることができる。

【0138】

監視・分析サーバ 1600 は、性能情報管理部 1602 及び分析レポート生成部 1603 によって構成される。性能情報管理部 1602 では、管理サーバ 105 から、ルールやルールの適用回数、レプリカ毎のリソース使用率等を、ルール・性能情報 1610 として取得する。得られたルール・性能情報 1610 は、ルール・性能情報データベース 1620 へ格納される。ルール・性能情報データベース 1620 は、時系列で情報を管理する。分析レポート生成部 1603 では、性能情報に基づいて分析レポート 1612 を生成する。

【0139】

また、監視・分析サーバ 1600 によって、管理サーバ 105 のリモート保守 1613 をする。

【0140】

情報システムの経営者又は運営者は、その監視分析の対価としてサービス料、保守料 1611 を MSP へ払う。情報システムの経営者又は運営者は、データベース管理 (又は、その一部) を外部委託することが可能となる。

【0141】

図 18 は、本発明の第 3 の実施の形態の監視・分析サーバ 1600 と管理サーバ 105 との間の処理のフローチャートを示す。図中左側が監視・分析サーバ 1600 の処理を、右側が管理サーバ 105 の処理である。

【0142】

監視・分析サーバ1600は、管理サーバ105へアクセス要求を送信する（ステップ1400）。

【0143】

管理サーバ105は、監視・分析サーバ1600からのアクセスを受け付け、認証を行う（ステップ1410）。認証が成功した場合は（ステップ1411）、ルールデータベース140に記録されるルール・性能情報を送信する（ステップ1412）。

【0144】

要求（1400）が認められた監視・分析サーバ1600は、ルール・性能情報1610を取得する（ステップ1401）。取得したルール・性能情報1610は、ルール・性能情報データベース1620に格納される。

【0145】

そして、ルール・性能情報データベース1620に格納される性能情報について、後述する方法によって時系列解析をして、分析レポートを生成する（ステップ1402）。最後に、生成した分析レポートをメールで送信する（ステップ1403）。なお、生成した分析レポートを監視・分析サーバ1600（又は、管理サーバ105）がネットワークを介してアクセス可能なwebサーバ（図示省略）に格納すると共に、分析レポートが作成されたことを管理サーバ105へ通知してもよい。

【0146】

そして、管理サーバ105は分析レポートを受信する（ステップ1413）。

【0147】

その後、監視・分析サーバ1600では、ルールに問題（例えば、ある特定のルールに問題があるためのスループットの低下）があるかを判定する（ステップ1404）。そして、問題となるルールを編集する（ステップ1405）。例えば、問題となるルールを削除したり、該ルールの相性値を変更することによって、ルールの適用を制御して、スループットを向上させる。そして、編集されたルールを管理サーバ105に送信する（ステップ1406）ことによってルールの保守を行う。

【0148】

管理サーバ105は、監視・分析サーバ1600から送信されたルールを受け付け、ルールデータベース140に記憶する（ステップ1414）。そして、ルールをフロントエンド102へ送信し（ステップ1415）、新しいルールでのスケジューリングを指示する。

【0149】

図19は、本発明の第3の実施の形態の監視・分析サーバ1600が、性能情報データベース1620において管理するデータファイルを説明する図である。

【0150】

データファイル2005は、ルール2000、2001及び性能情報2002によって構成される。ルールは、ルールリスト1（2000）及びルールリスト2（2001）によって構成される。ルールリスト1（2000）及び性能情報（2002）は、各々レプリカ104の数だけ存在する。

【0151】

また、データファイル2005は、ルールが変更される度に新たに生成され、時系列データとして管理される。

【0152】

図20は、本発明の第3の実施の形態の分析レポートの表示画面の例を説明する図である。

【0153】

分析レポート1500として、スループット表示1501、ルール表示1502、性能表示1503が表示されている。

【0154】

スループット表示1501としては、システム全体の性能のグラフが表示される。例えば、単位時間あたりの問い合わせ処理量を表示する。スループット表示1501によって、システム全体の性能の時系列変化を把握することができる。また、グラフを外挿することによって、将来のスループットを予測することもできる。

【0155】

ルール表示 1 5 0 2 では、例えば、ある時間にフロントエンドにおいてスケジューリングに使用されたルールに関する情報を表示する。ルールを解析する期間（時間）は、ユーザが直接入力してもよいし、スループット画面 1 5 0 1 で指定してもよい。ルール表示 1 5 0 2 では、ルールの条件部、相性値、適用回数が表示される。ルール表示 1 5 0 2 でルールの条件部に表示されるのは問い合わせにつけられた I D であり、問い合わせの内容ではないため、ルール詳細表示 1 5 1 0 を別に設け、実際のルールの内容を表示する。ルール表示 1 5 0 4 によって、どのようなルールが生成され、それぞれ何回適用されているのかを把握することができる。

【 0 1 5 6 】

また、適用されるルールの時系列的な変化を知ることができる。例えば、ルールは学習され変化していくため、スループットが低下している場合は、適用されるルールに変化があるかどうかをチェックすることによって、スループット低下の原因を調査することができる。ある特定のルールに問題がある場合は、ルールを編集することによって保守を行う。例えば、該ルールを削除したり、該ルールの相性値を変更することによって、ルールの適用を制御して、スループットを向上させることができる。

【 0 1 5 7 】

性能画面 1 5 0 3 は、レプリカ毎に C P U 利用率、ディスク利用率といった性能リソースに関する情報をグラフ表示する。性能画面によってレプリカ毎の負荷や、レプリカ間の負荷バランスを把握することができる。この性能画面によって、レプリカ毎のボトルネックが分かれば、そのボトルネックを回避するための設備投資提案をすることが可能である。例えば、メモリの使用率が高くなっていれば、メモリ増設を提案することができる。

【 0 1 5 8 】

図 2 1 は、第 3 の実施の形態による監視、管理サービスのビジネスモデルの概略を説明する図である。

【 0 1 5 9 】

データベースシステム（例えば、図 1）等の I T システム 1 8 0 0 のシステム

の管理者又は経営者は、外部のMSP 1801と契約を結び管理をアウトソーシングする。

【0160】

ITシステム1800からは、ルール・性能情報1802をMSP 1801へ送信する。このルール・性能情報1802は、サーバのリソース使用率といった性能情報に加え、ITシステム1800において自律的に生成されたルールも含む。

【0161】

MSP 1801では、従来行われてきたリソース使用率等の性能情報の監視・分析に加え、ルールの監視・分析を実施する。MSP 1801では、ルール・性能情報1802を受けて、分析を実施して分析結果をレポート1803にまとめ、ITシステム1800へ提供する。また、MSP 1801では、リモート保守1805を行う。ここでは、従来のソフトウェアの設定パラメータの変更等に加え、ルールの編集を行う。ルールは、複数のデータを抽象化して（例えば、if-then形式で）記載されており、可読性が高いため人間による理解も容易であることから、MSP 1801の管理者によって編集をすることができる。

【0162】

また、MSP 1801による診断の上、レプリカ104等のハードウェアの変更提案も可能である。ルール・性能情報1802を時系列で管理することによって、システムの変化を把握すれば、性能上の問題が発生する前に対策を取ることができる。

【0163】

ITシステム1800の管理者又は経営者は、保守やサービスに対し対価1804をMSP 1801に支払う。

【0164】

より具体的には、MSP 1801では、分析レポート1803を定期的に実行する。ユーザである、ITシステム1800の管理者又は経営者は、その対価として、サポート・保守料をMSP 1801に支払う。

【0165】

以上説明したように、第3の実施の形態では、外部の監視・分析サーバ1600によって、管理サーバ105の監視・管理をするので、抽象化されたルールを扱うことによって、システムの挙動の把握が容易であり、ルール変更による保守も容易であるため、高付加価値サービスの提供が可能である。

【図面の簡単な説明】

【図1】

本発明の第1の実施の形態のデータベースシステムの構成図である。

【図2】

本発明の第1の実施の形態のレプリカ104の処理のフローチャートである。

【図3】

本発明の第1の実施の形態のレプリカ104で記録されるログデータの構成図である。

【図4】

本発明の第1の実施の形態の管理サーバ105の処理のフローチャートである。

【図5】

本発明の第1の実施の形態の問い合わせのグループ化の説明図である。

【図6】

本発明の第1の実施の形態の問い合わせの実行状態の説明図である。

【図7】

本発明の第1の実施の形態において相性計算に用いられる相性マトリックスの説明図である。

【図8】

本発明の第1の実施の形態の相性計算（ステップ502）のフローチャートである。

【図9】

本発明の第1の実施の形態の問い合わせ間のルールの生成（ステップ503）のフローチャートである。

【図10】

本発明の第 1 の実施の形態のルールリストの構成図である。

【図 11】

本発明の第 1 の実施の形態において相性計算に用いられる別の相性マトリックスの説明図である。

【図 12】

本発明の第 1 の実施の形態の管理サーバ 105 がルールデータベース 140 において管理するデータファイルの説明図である。

【図 13】

本発明の第 1 の実施の形態のフロントエンド 102 の処理のフローチャートである。

【図 14】

本発明の第 1 の実施の形態のスケジューリング（ステップ 604）のフローチャートである。

【図 15】

本発明の第 2 の実施の形態において相性計算に用いられるユーザ間の相性マトリックスの説明図である。

【図 16】

本発明の第 2 の実施の形態のスケジューリング（ステップ 604）のフローチャートである。

【図 17】

本発明の第 3 の実施の形態のデータベースシステムの構成図である。

【図 18】

本発明の第 3 の実施の形態の監視・分析サーバ 1600 と管理サーバ 105 との間の処理のフローチャートである。

【図 19】

本発明の第 3 の実施の形態の監視・分析サーバ 1600 が、性能情報データベース 1620 において管理するデータファイルの説明図である。

【図 20】

本発明の第 3 の実施の形態の分析レポートの表示画面の例を説明する図である。

【図 2 1】

監視、管理サービスのビジネスモデルの概略図である。

【図 2 2】

従来のデータベースシステムの構成図である。

【図 2 3】

従来のデータベースシステムのデータベースバッファの動作の説明図である。

【図 2 4】

従来のデータベースシステムのデータベースバッファの動作の説明図である。

【符号の説明】

- 100 クライアント
- 101 ネットワーク
- 102 データベースサーバ（フロントエンド）
- 103 データベースサーバ（マスタ）
- 104 データベースサーバ（レプリカ）
- 105 管理サーバ
- 106 マスタディスク
- 107 レプリカディスク
- 110 ログ記録・送信部
- 111 問い合わせ実行部
- 120 ログ取得部
- 121 相性計算部
- 122 ルール生成部
- 123 ルール管理部
- 140 ルールデータベース
- 200 サーバ
- 201 ディスク
- 203 メモリ
- 1600 監視・分析サーバ

1 6 0 1 ネットワーク

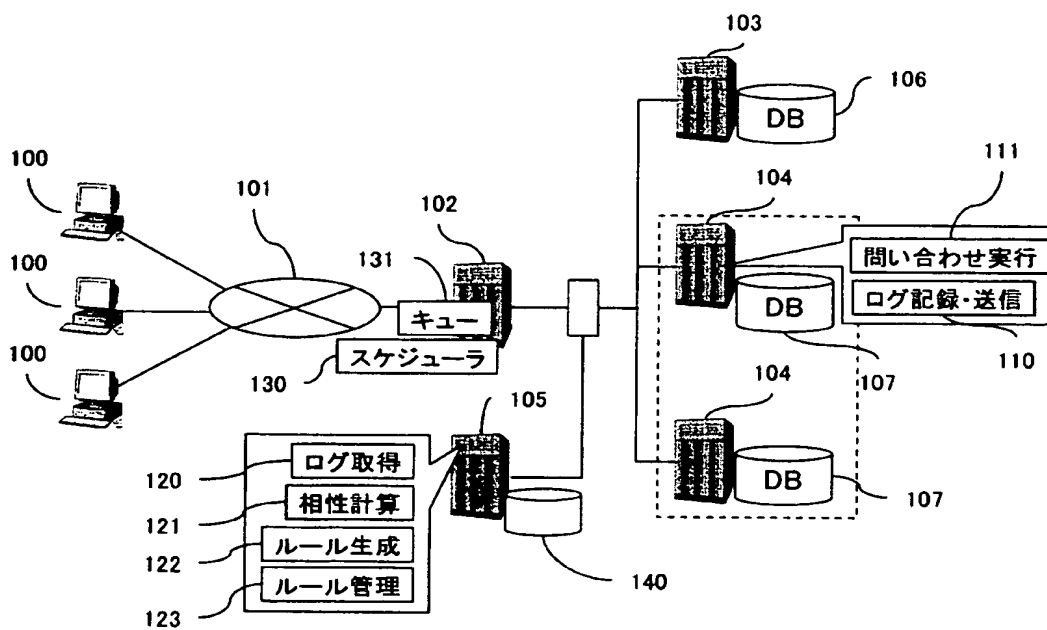
1 6 0 2 性能情報管理部

1 6 0 3 分析レポート生成部

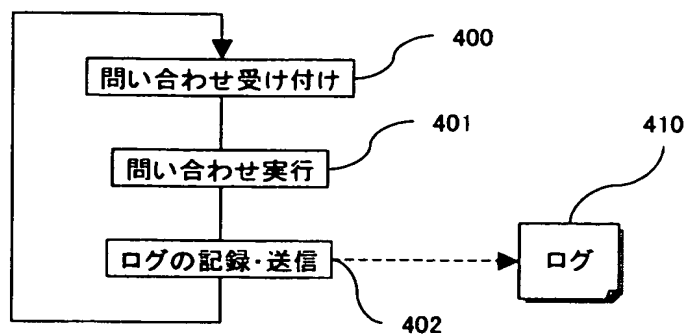
【書類名】

図面

【図 1】



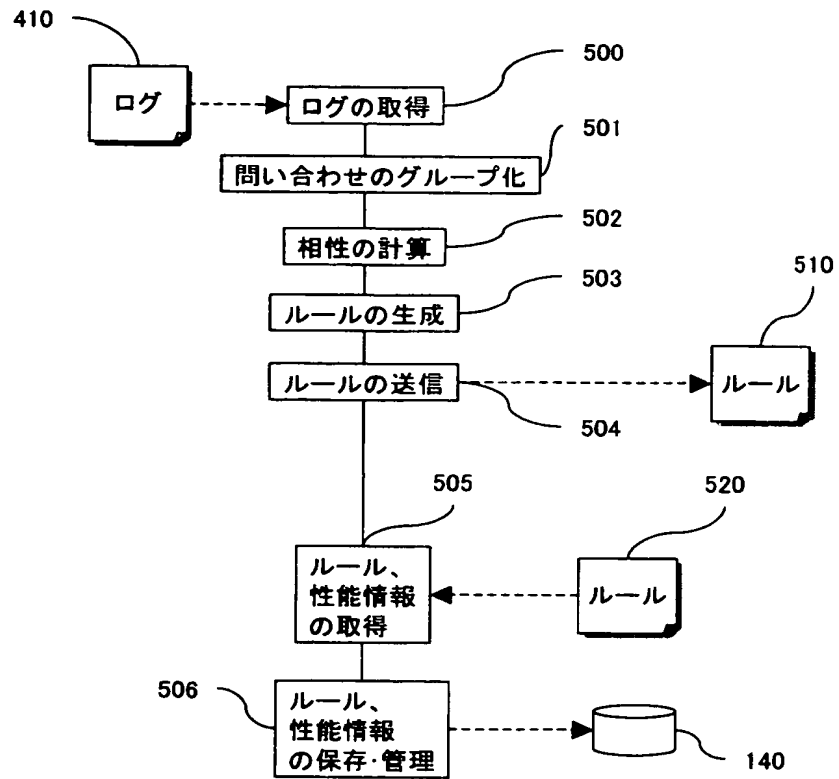
【図 2】



【図 3】

		700	701	702	703	704
710	720	問い合わせ内容	処理時間	開始時刻	終了時刻	ユーザ名
		select ...	2200	19:05 22:02	19:05 24:22	Scott
		select ...	1500	19:05 25:00	19:05 26:50	John

【図 4】



【図 5】

(A) グループ化例1

1050		1051	
#	テンプレート	問い合わせ内容	
1	A	select a from A where b>2	1011
		select a,b from A	1012
2	B	select a from B where c>3	1013
		select a,c from B where b>3	1014

1010

(B) グループ化例2

1050		1051	
#	テンプレート	問い合わせ内容	
1	{a},{A}	select a from A where b>2	1021
		select a from A	1022
2	{a,c},{A}	select a,c from A where c>3	1023
		select a,c from A where d>3	1024

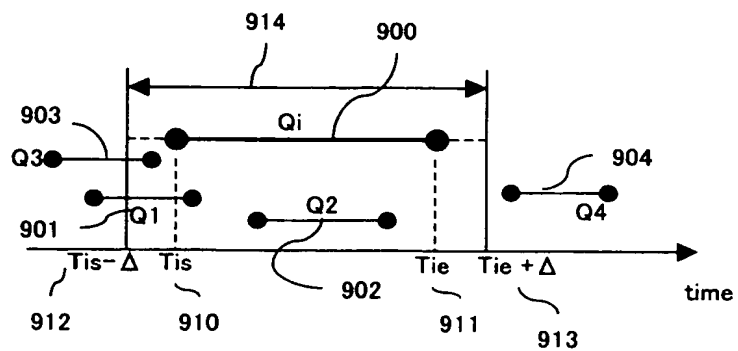
1020

(C) グループ化例3

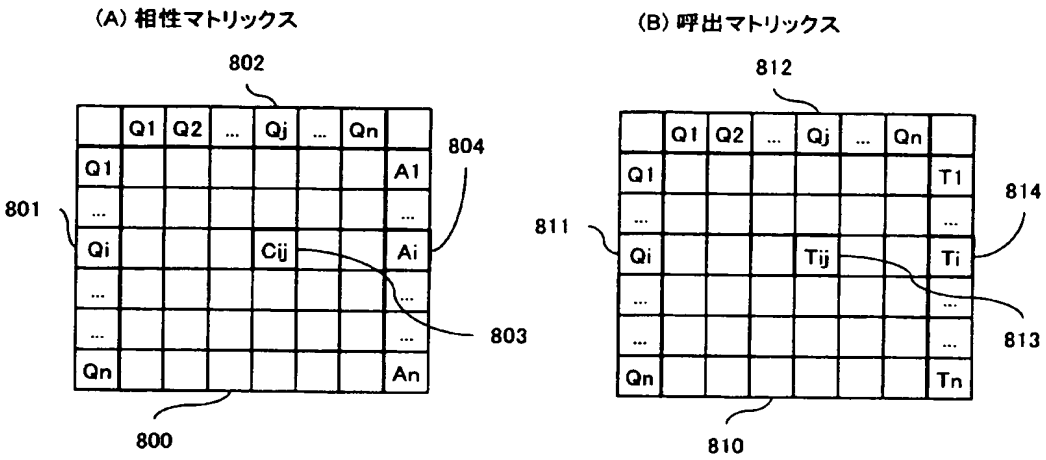
1050		1051	
#	テンプレート	問い合わせ内容	
1	{a},{A},{b}	select a from A where b>2	1031
		select a from A where b >30	1032
2	{a},{A},{c}	select a from A where c>3	1033
		select a from A where c>5	1034

1030

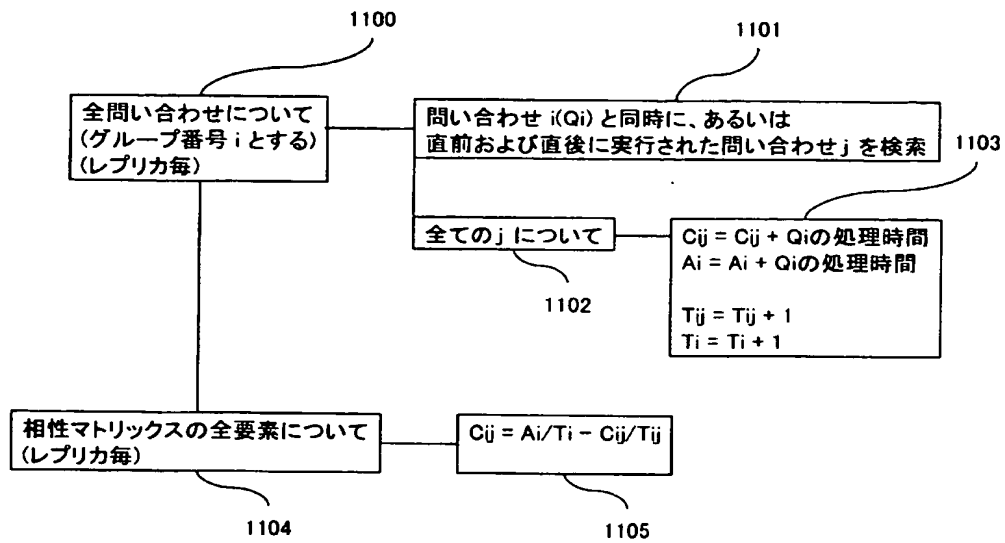
【図 6】



【図 7】

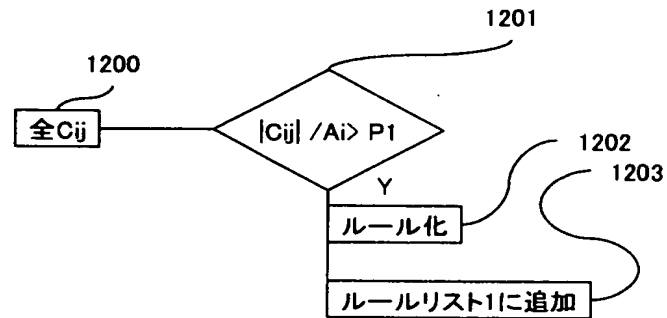


【図 8】

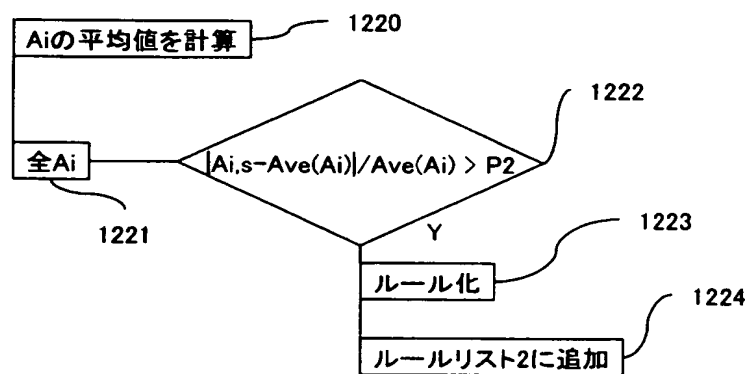


【図 9】

(A) 問い合わせ間の相性のルール化



(B) 問い合わせとサーバの相性のルール化



【図 10】

(A) ルールリスト1(問い合わせ間の相性のルール)

1914

1911 1912 1913

条件	相性値	回数
Q1,Q2	20	0
Q1,Q7	10	0
...

1910

(B) ルールリスト2(問い合わせとサーバの相性のルール)

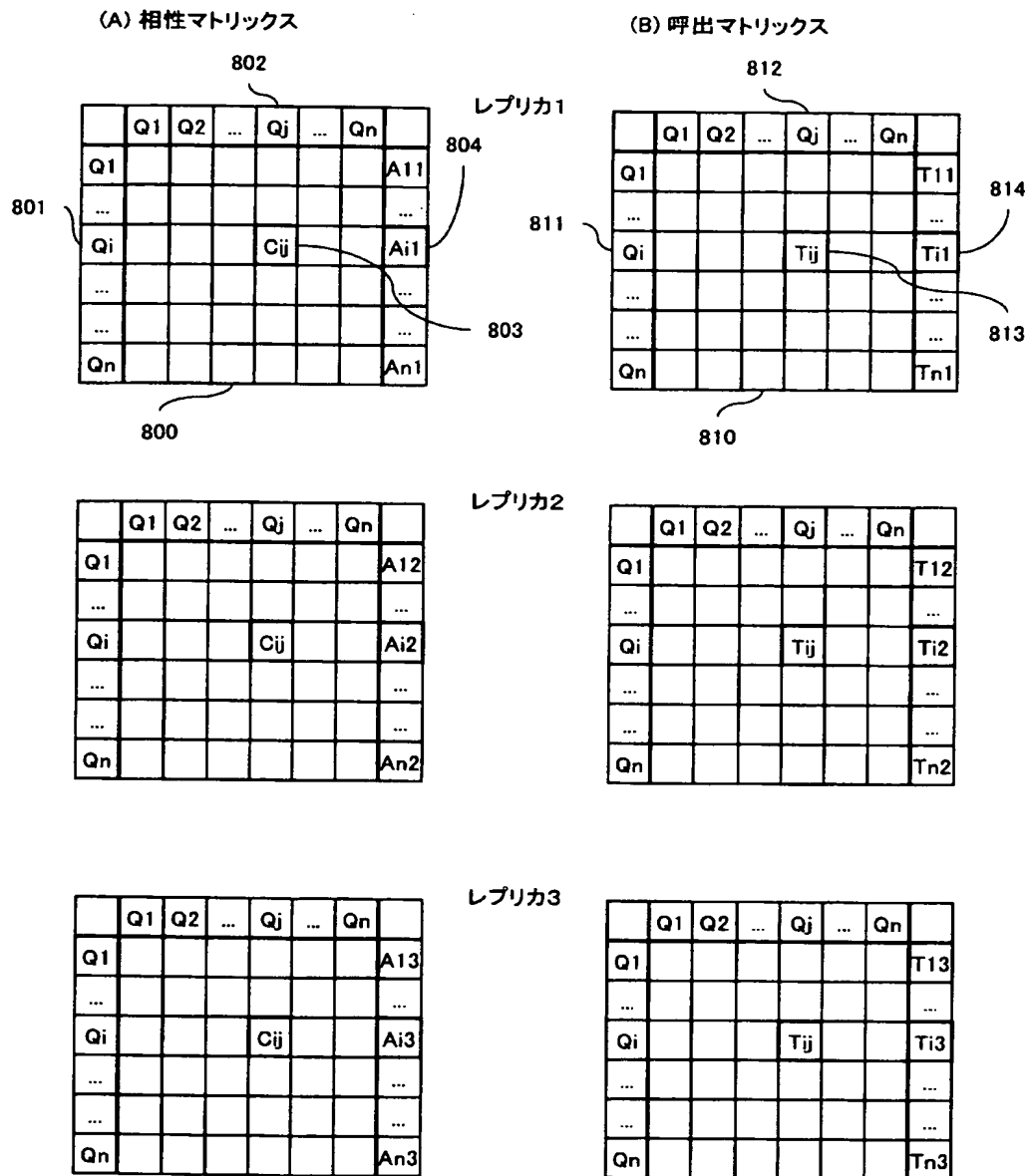
1934

1931 1932 1933

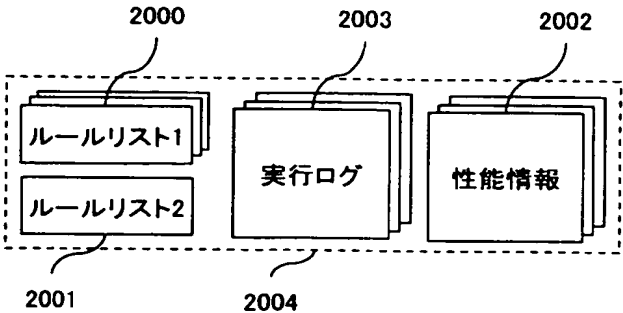
条件	相性値	回数
Q1,S2	20	0
Q1,S4	-10	0
...

1930

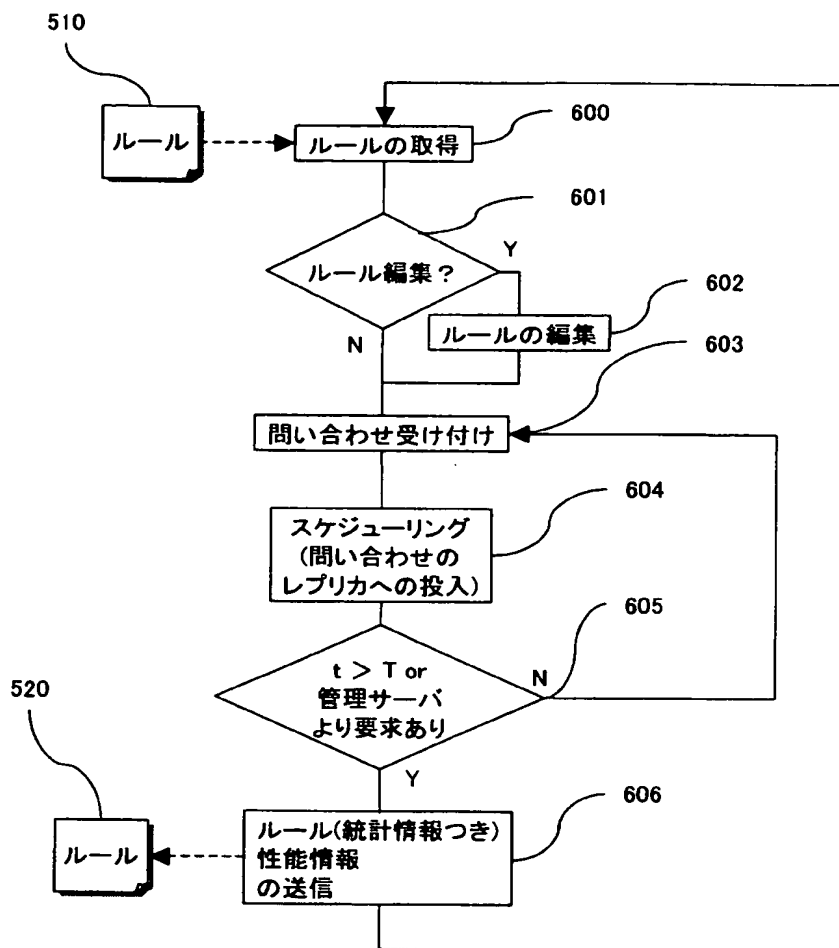
【図 11】



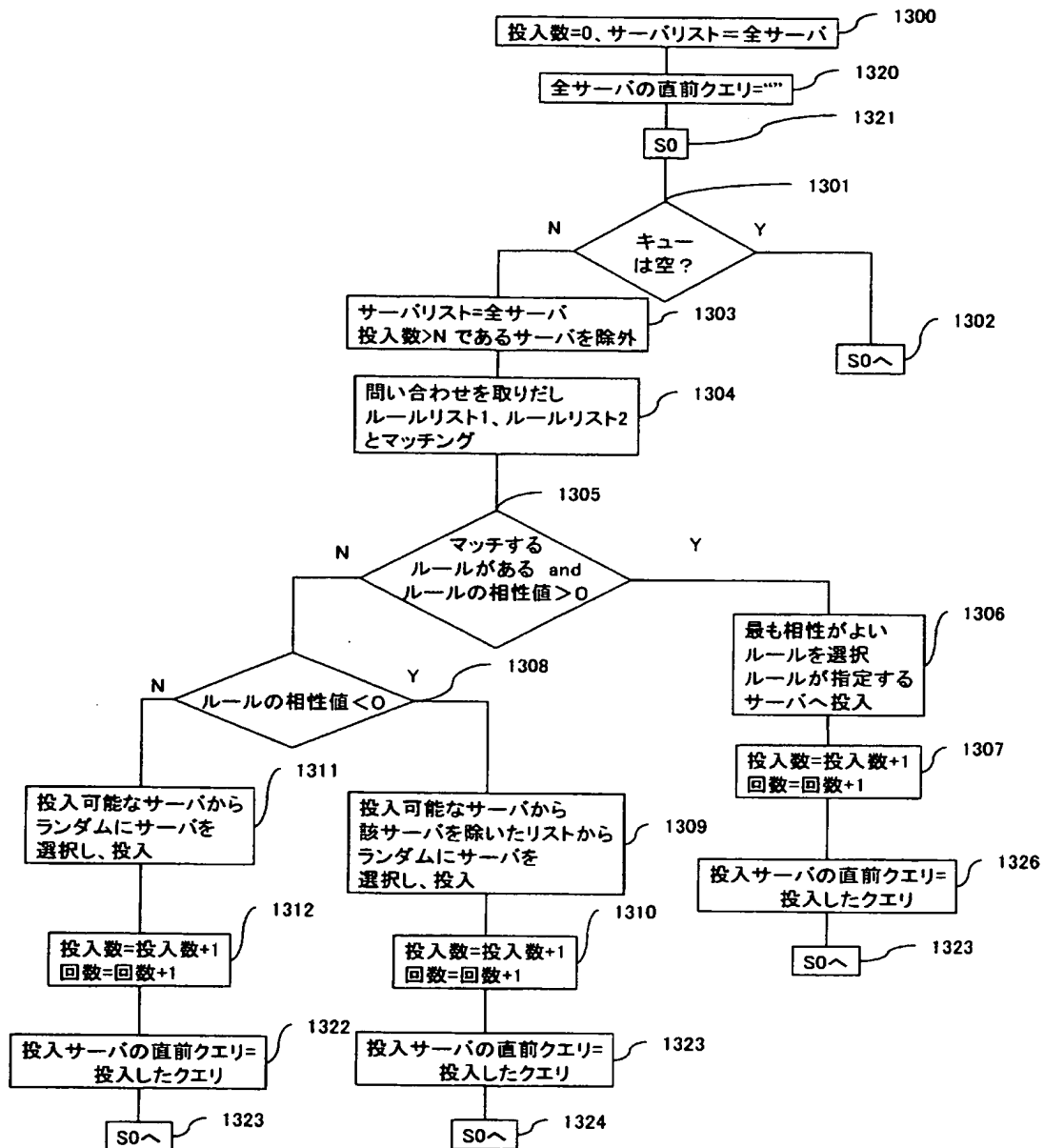
【図 12】



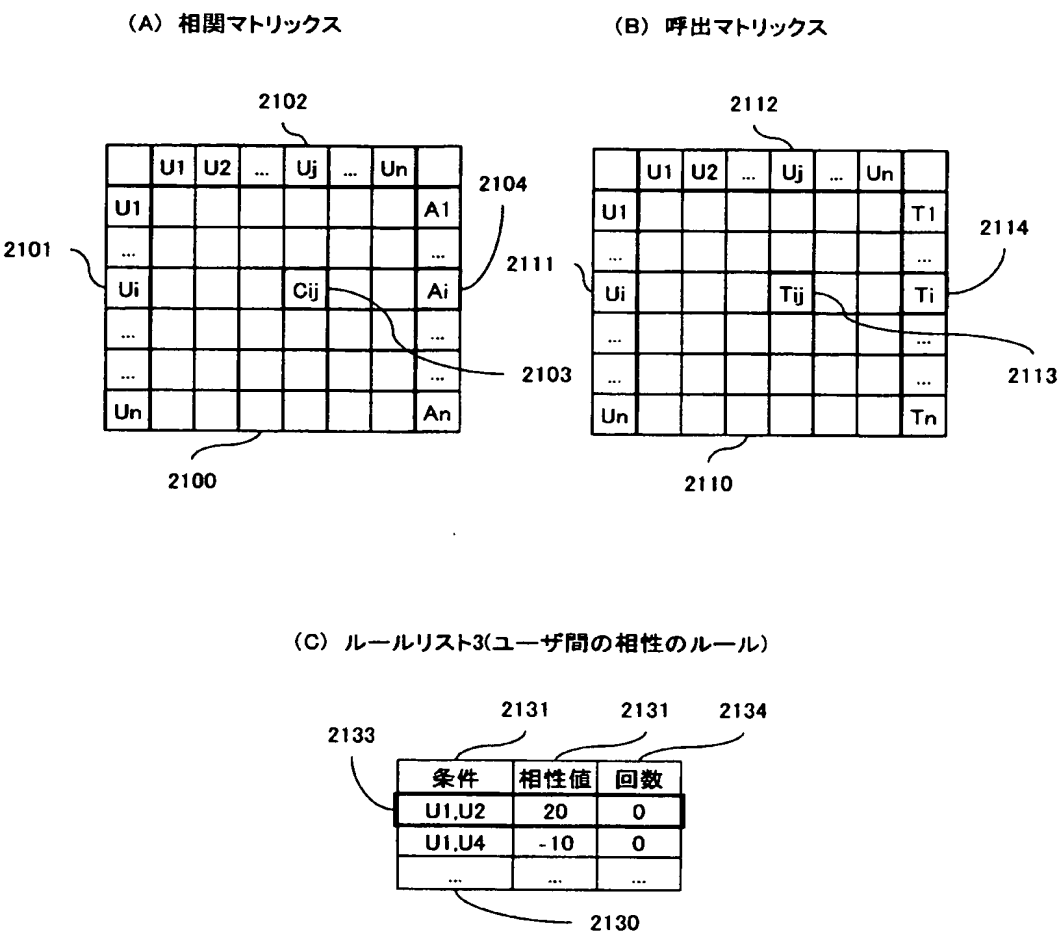
【図 13】



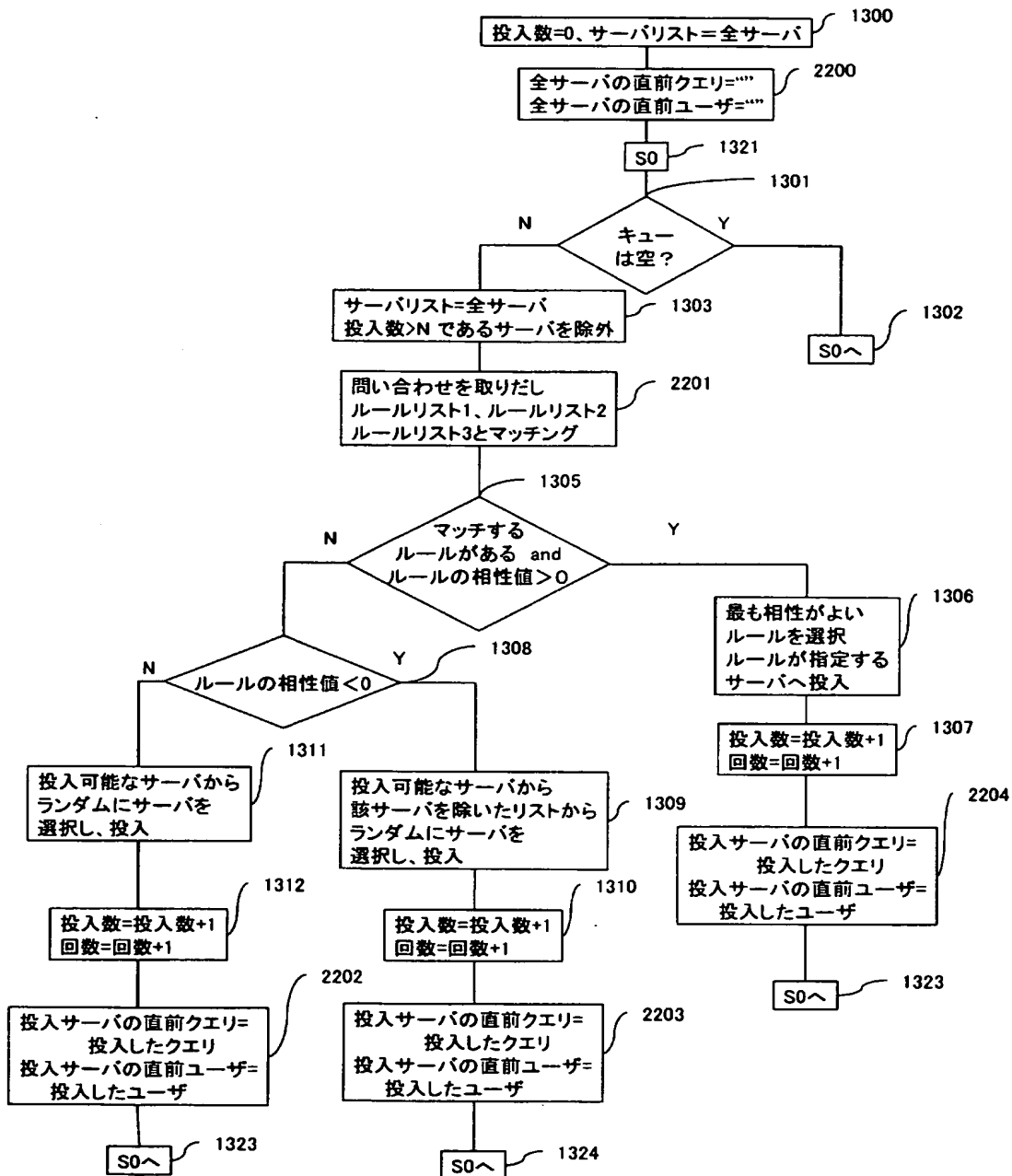
【図 14】



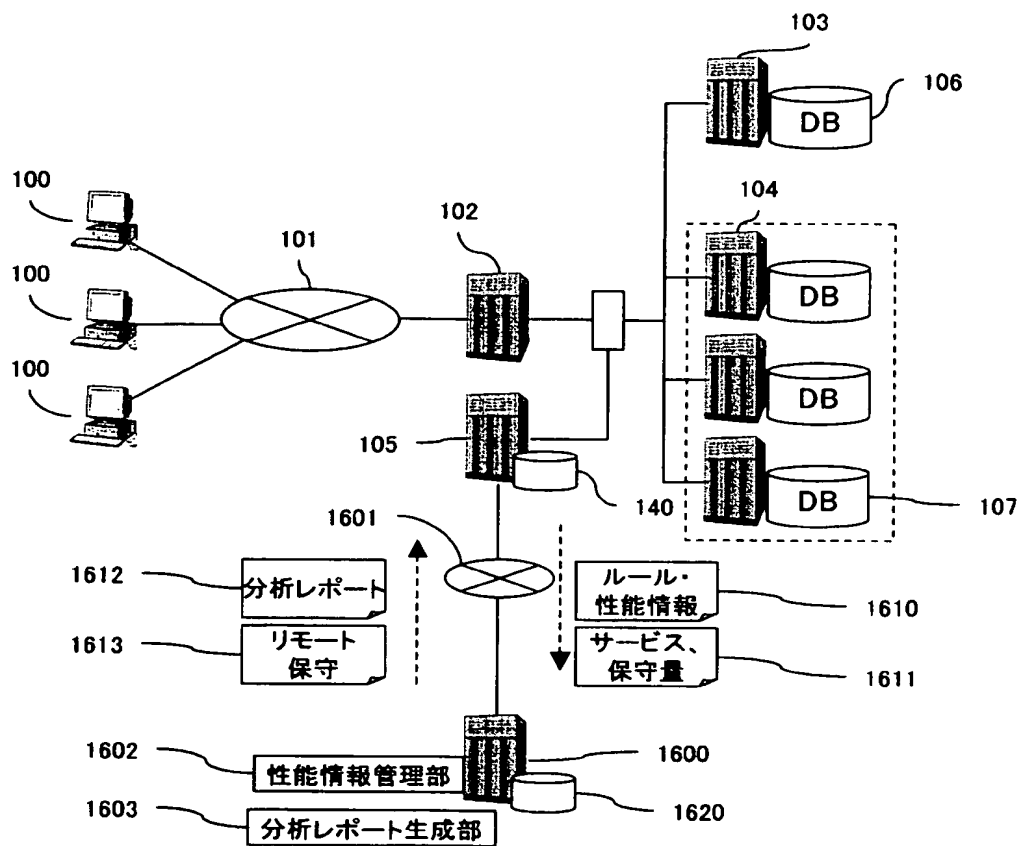
【図 15】



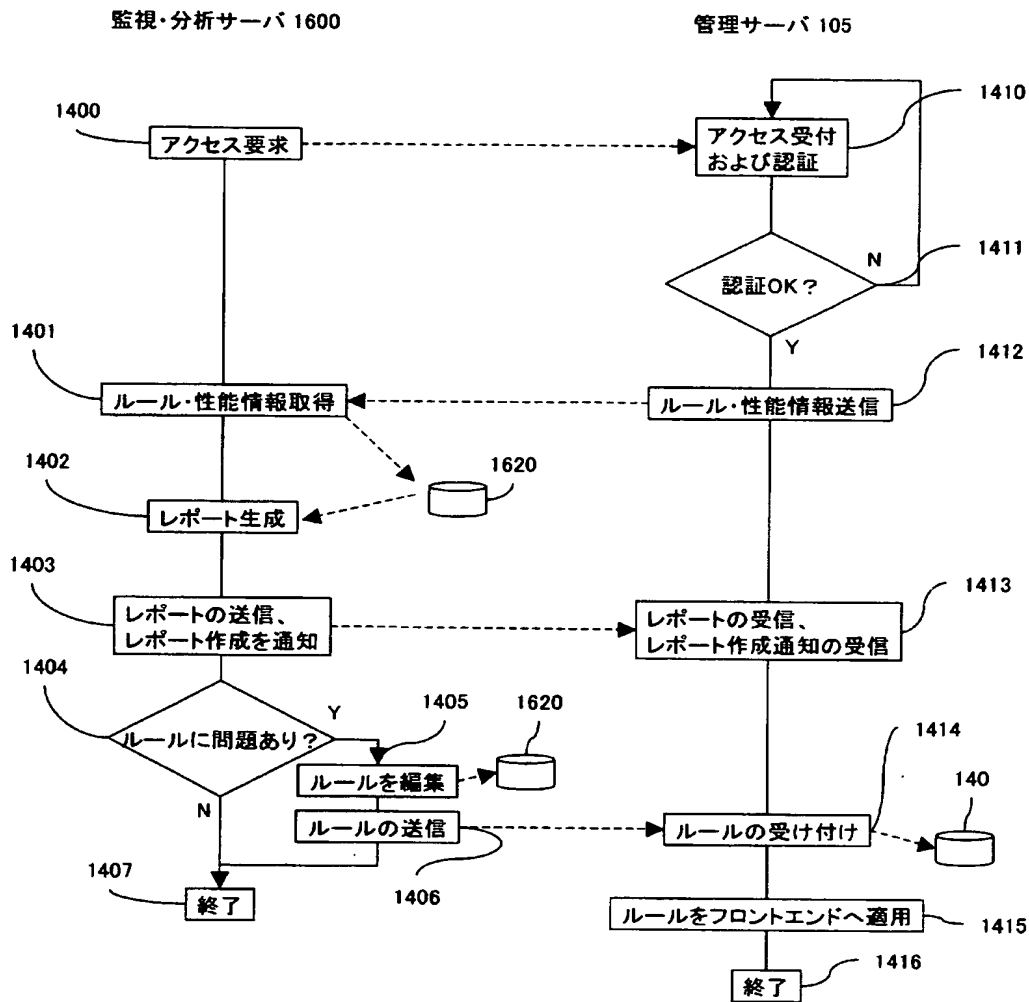
【図 16】



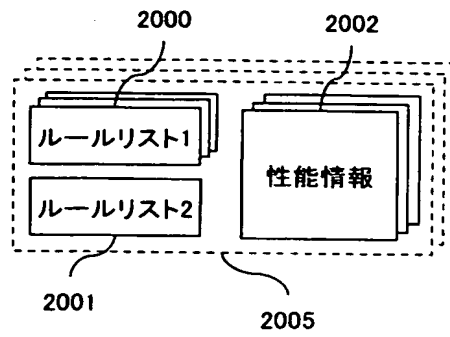
【图 17】



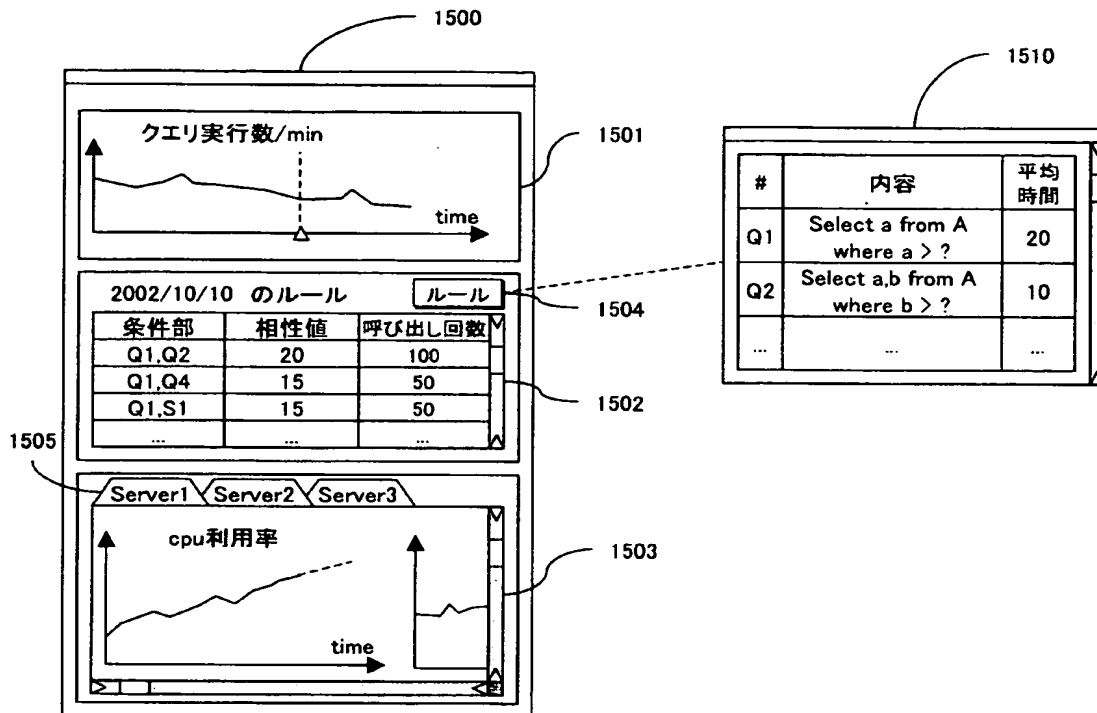
【図18】



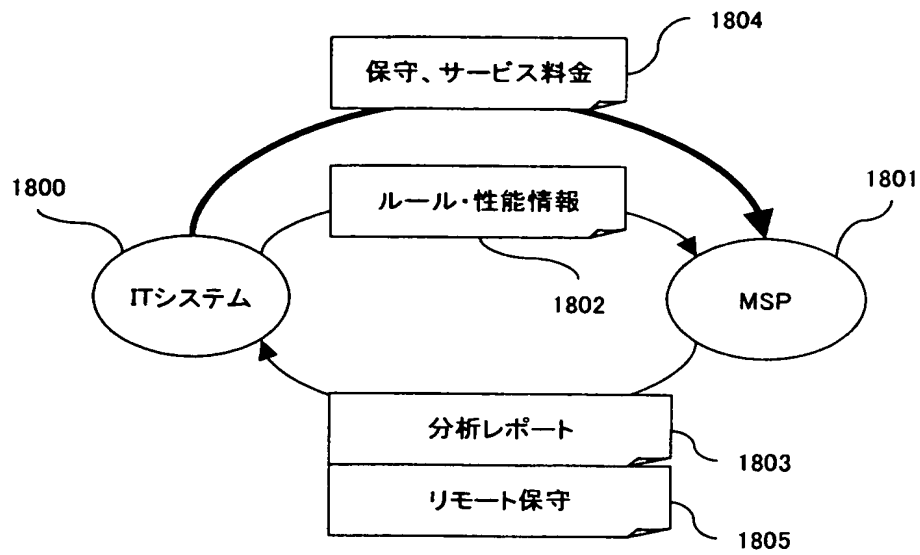
【図 19】



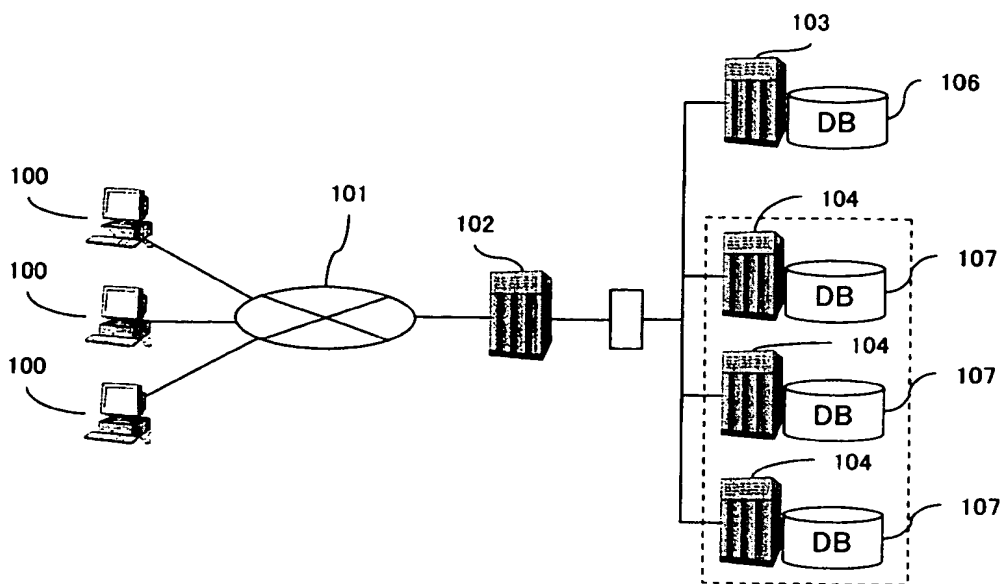
【図 20】



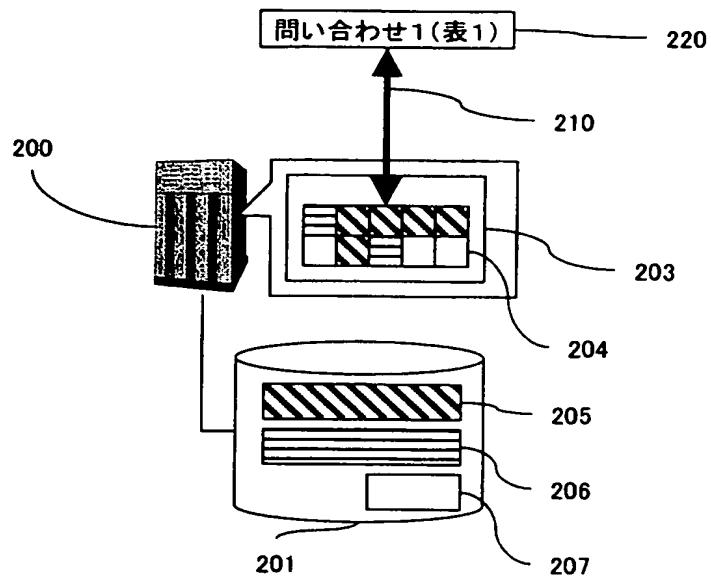
【図 21】



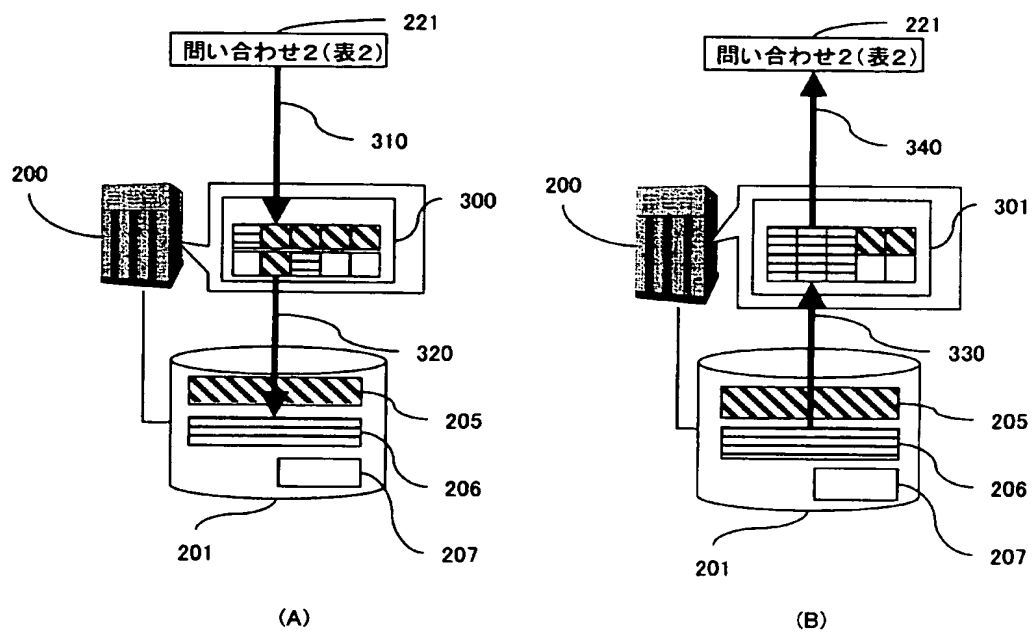
【図 22】



【図 23】



【図 24】



【書類名】 要約書**【要約】**

【課題】 並列・分散型データベースシステムにおいて、リソース競合を回避し、システム全体の性能を向上させる。

【解決手段】 同一内容の検索が可能なデータベースを有し、問い合わせ要求に従って該データベースを検索する複数のデータベースサーバに対して、定められたルールを用いて問い合わせを投入するフロントエンドサーバと、前記フロントエンドサーバが使用するルールを管理する管理サーバと、を備えるデータベースシステムで用いられる問い合わせ投入方法において、前記管理サーバは、前記データベースサーバの実行ログを取得し、前記取得した実行ログを用いて計算された問い合わせに関する相性に基づいて前記ルールを生成し、前記フロントエンドサーバは、前記管理サーバで生成されたルールを用いて、前記問い合わせを投入することを特徴とする。

【選択図】 図 1

特願 2003-071908

出願人履歴情報

識別番号

[000005108]

1. 変更年月日

1990年 8月31日

[変更理由]

新規登録

住 所

東京都千代田区神田駿河台4丁目6番地

氏 名

株式会社日立製作所